
Black Fennec

Release 0.12.0

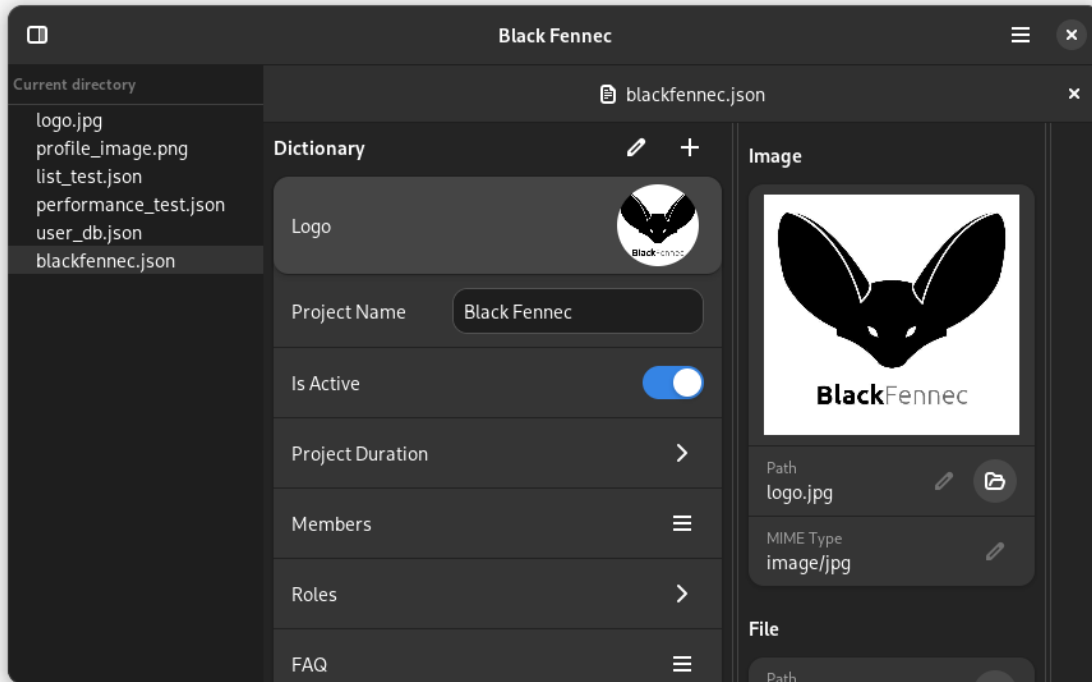
blackfennec.org

Dec 23, 2022

CONTENTS

1 Usage	3
2 Development	5
3 Project	7
3.1 Usage	7
3.2 Development	9
3.3 Project	214
3.4 Index	279
Python Module Index	281
Index	287

Black Fennec is free and open source software. It is a simple and easy to use application for viewing and editing semi-structured data like JSON in a visually interpreted form. It is written in Python and uses GTK4 and Libadwaita for the user interface.



You can find the source code on [Gitlab](#) or as a read-only mirror on [GitHub](#). If you want to submit any issues or feature requests, please use the [issue tracker](#) on GitHub.

To find out more about this project refer to the following pages.

USAGE

Here you will learn how to install and use Black Fennec. This section is intended for users who want to use Black Fennec and somehow stumbled upon this documentation.

If you are of the impatient type and just want to get started, just [download Black Fennec via FlatHub](#).

DEVELOPMENT

This section is intended for developers who want to contribute to Black Fennec. It contains information about the *architecture of the application* and the *domain model*, how to build it and how to contribute to the project.

Additionally it provides a guide on *how to write extensions* for Black Fennec.

Alternatively, you can install Black Fennec by downloading the `.flatpakref` file from the FlatHub website. This file, when opened, should give you the option to install Black Fennec.

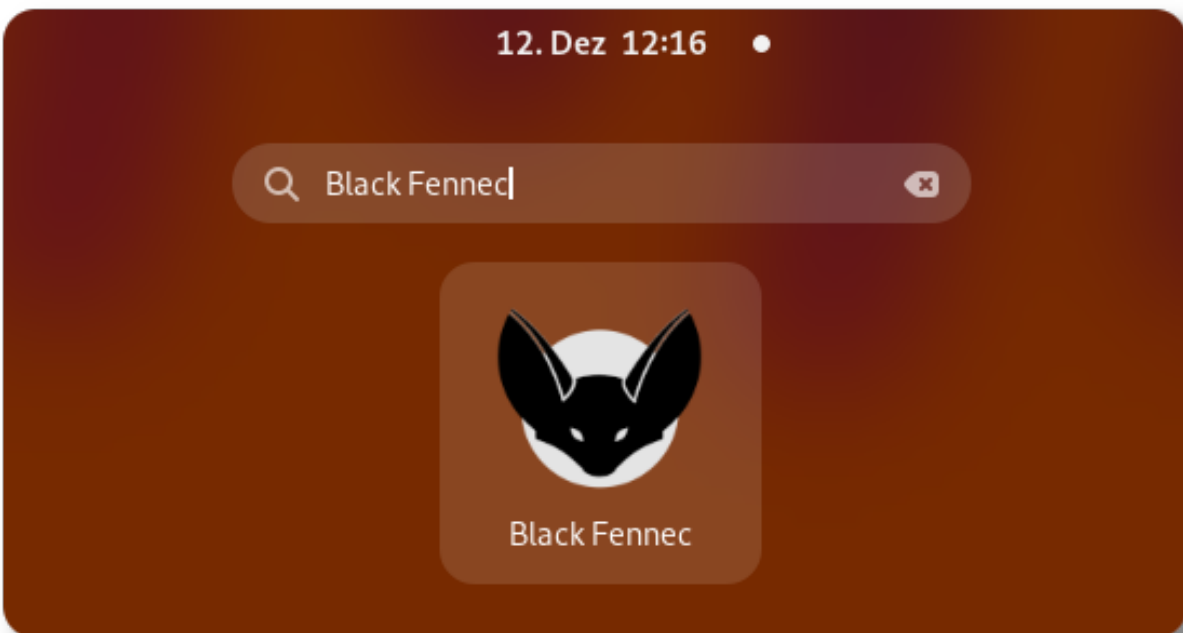
Extensions

Black Fennec can be enhanced with extensions. It is possible to develop your own extensions. An introduction for which can be found in the [Extension Development](#) section of this documentation. The *core* and *base* extensions should be installed by default. In fact, Black Fennec will warn you if they are not installed.

Extensions can be installed through your software manager of choice. In the case of GNOME Software, you should see a list of available extensions just below the application description in a section called *Add-ons*. Installing an extension is as simple as clicking the checkbox.

3.1.2 Running Black Fennec

Once you have installed Black Fennec, you can run it by searching for it in your application launcher. In GNOME, you can search for Black Fennec in the Activities Overview.



Black Fennec starting with version 0.11 also registers for the mime type *application/json*. This means that you can open JSON files with Black Fennec by double clicking them in your file manager. If you have another application installed to handle JSON you might need to use the *open with* option.

3.1.3 Command Line

If you want to install Black Fennec using the command line you can use the command below if Flatpak and FlatHub are installed correctly.

```
flatpak install flathub org.blackfennec.app
```

Running Black Fennec from the command line is possible by using the command below.

```
flatpak run org.blackfennec.app
```

Installing extensions is also possible from the command line. The command below installs the *core* extension.

```
flatpak install flathub org.blackfennec.app.extension.core
```

3.2 Development

Black Fennec is, in essence, a glorified JSON viewer with a powerful extension API. Extensions can add custom types, views, actions and even new mime types. As such this documentation caters to two types of developers.

The goal of Black Fennec development is to ensure the stability, extensibility and usability of the software. This is achieved by following a set of standards and best practices which are described in *project standards*. As a developer you should be aware of these standards. However, it is also important to understand the bigger picture, the architecture and the domain model. This is explained in *architecture* and *domain model* respectively.

Extension developers make Black Fennec more than just a glorified JSON viewer. The available extensions largely determine the usefulness of Black Fennec. Thus a *dedicated section* explains how to write the different types of extensions.

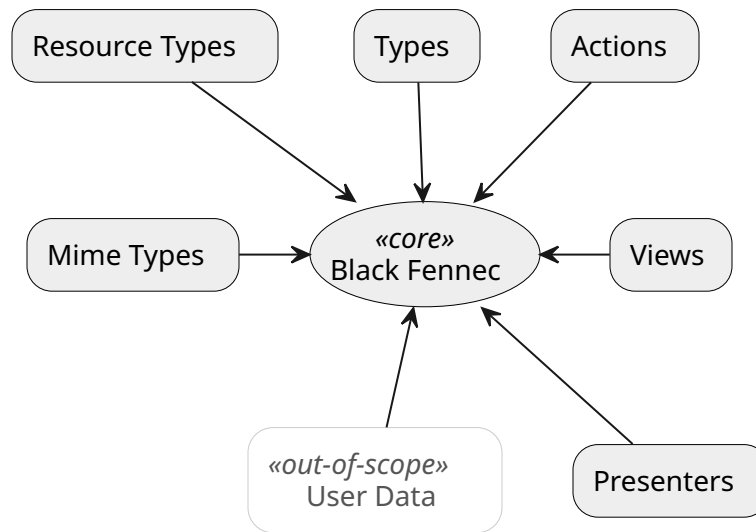
In the *testing* section we record and document system and usability tests.

Finally, the *code* section contains a generated source code documentation of Black Fennec.

3.2.1 Architecture

The ecosystem of Black Fennec can be divided into four components and the means by which they interact. From the architectural standpoint, Black Fennec provides the interfaces between the components and also some services to solve common problems. These interfaces and services are largely independent and allow for the high degree of cohesion and low coupling which is required to manage the complexity in this project. The following graph gives a good overview. It does not strive to be a perfect reflection of the code and its complexity but instead should serve as an entry point for new developers.

Architecture Overview



Black Fennec is a desktop application without backend integration. All code runs within a single tire. However, the project entails a fair amount of engineering challenges.

User Data

User data is the information the user is viewing/editing with Black Fennec. As such it is not part of the source code. When it's loaded from a file into Black Fennec it is deserialized into the *object model*. The resulting structure will be interpreted through the type system, visualized partially by extension provided views, and displayed by a presenter.

Types

Types can be added by extensions. The richer the ecosystem and the more available types the better the interpretation. At least that's the theory.

Actions

Actions are extension defined procedures which the user can trigger. They can define preconditions in the form of a type and are able to run code and manipulate the structure.

Resource Types

A *Resource Type* enables access to a specific type of resource. It is registered to a protocol and provides the implementation to read and write to it. It is used by the *document system* to abstract the access to the resource.

Mime Types

A *Mime Type* is responsible for parsing the contents of a *Resource* and is identified by a string. In combination it allows the *document system* to abstract the access to the resource and mitigates the combinatorial explosion of possible file formats and protocols.

Views

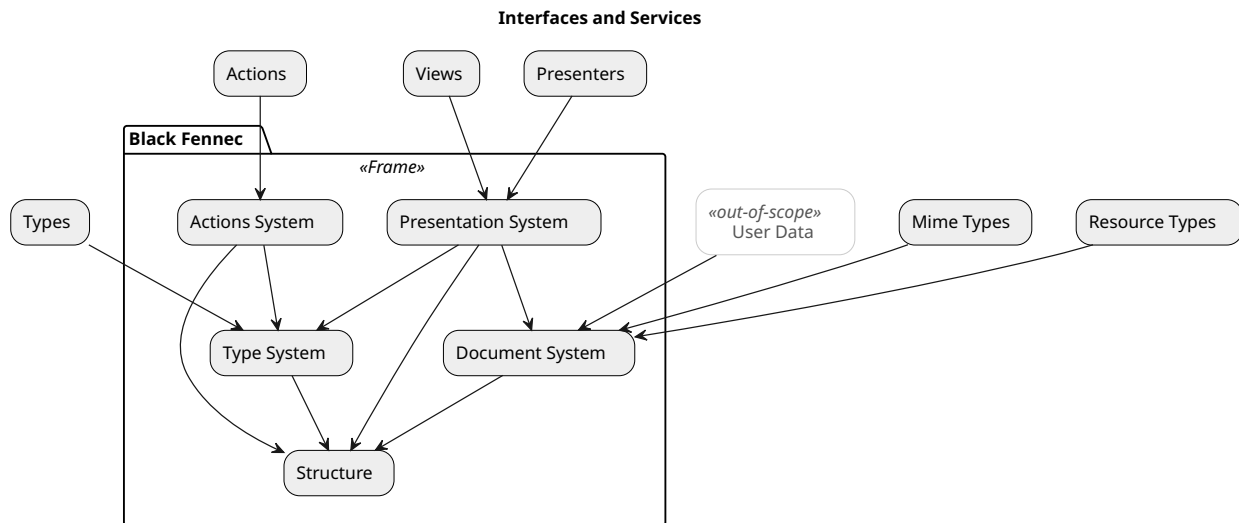
Visualisations are the visual representation of the Structure. They play a big role in what the user sees and interacts with. Internally, MVVM is used to decouple view and logic. Note that this - from the point of view of Black Fennec - is an implementation detail and neither enforced nor required by extensions.

Presenters

The presentation is the responsibility of the *presenter*. It displays the interpreted structure which are requested from the *interpretation service*. The presenter also observes the *navigation service* for navigation request and is responsible for acting on them.

Black Fenec

When we take a closer look at the interfaces and services used by the “external” components, we can identify some concepts that can be found throughout this documentation. These are the high-level concepts which are important to understand the bigger picture. Each component is conceptually largely independent of the others. This allows us to tune their view of the system to their needs. Think of the facade pattern but towards the core and not to hide legacy code and thus minimizes complexity whereby development efficiency is maximized.



Structure

The *structure* is the parsed user data. It is the foundation upon which the *interpretation* is built and the fabric in which navigation is performed. It is implemented in our *object model* and can be further preprocessed by *layers* such as the *overlay*. The layers allow specialized usage of the structure without manipulating the underlying data (*underlay*).

Type System

The *type system* is represented as a collection of known types that can be used to interpret the structure. They are stored in a registry. This allows runtime loading and unloading of the available types and is an important enabler of the extension infrastructure. The *interpretation service* does most of the heavy lifting as it decides which types from the *type system* ought to be used to visualize a given structure. The service can be configured on a ‘per request’ basis with a specification, giving fine-tuned control to the user of the service.

Action System

The *action system* is responsible to execute actions. It is the interface between the user and the *action* and is responsible to check the preconditions and to execute the action. It is also responsible to provide the user with a list of available actions.

Document System

The *document system* is responsible to import a structure from sources such as files. With that it is responsible to abstract the access protocol and mime type, both of which can be loaded at runtime by extensions.

Presentation System

The *presentation system* manages the available *views* and *presenters*. It hides all the complexity and ought to provide an intuitive interface for all possible interactions with the system.

Further information and more detailed descriptions of the mentioned components can be found in the *domain model*. If you are interested in the documentation of the source code [follow this link](#)

Extensions

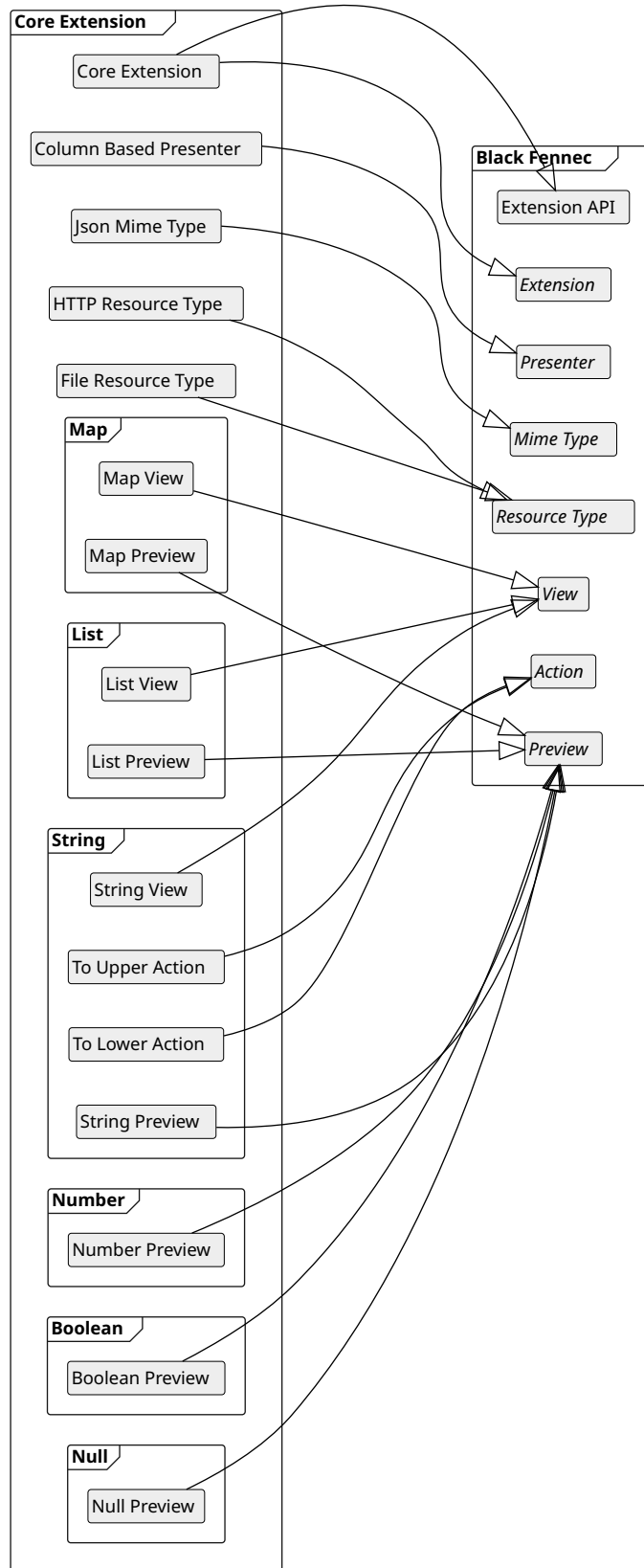
Extensions are a core part of Black Fennec but conceptually decoupled from the domain model through the extension api. Extensions are implemented as a set of classes which are loaded into the domain model at runtime. The extension api is a set of interfaces which extension can use to register functionality with Black Fennec.

As references for a working model, the following diagrams show the structure of the core and base extensions.

Core

The core extension provides a UI for the *structure*, some *actions*, as well as a *presenter*.

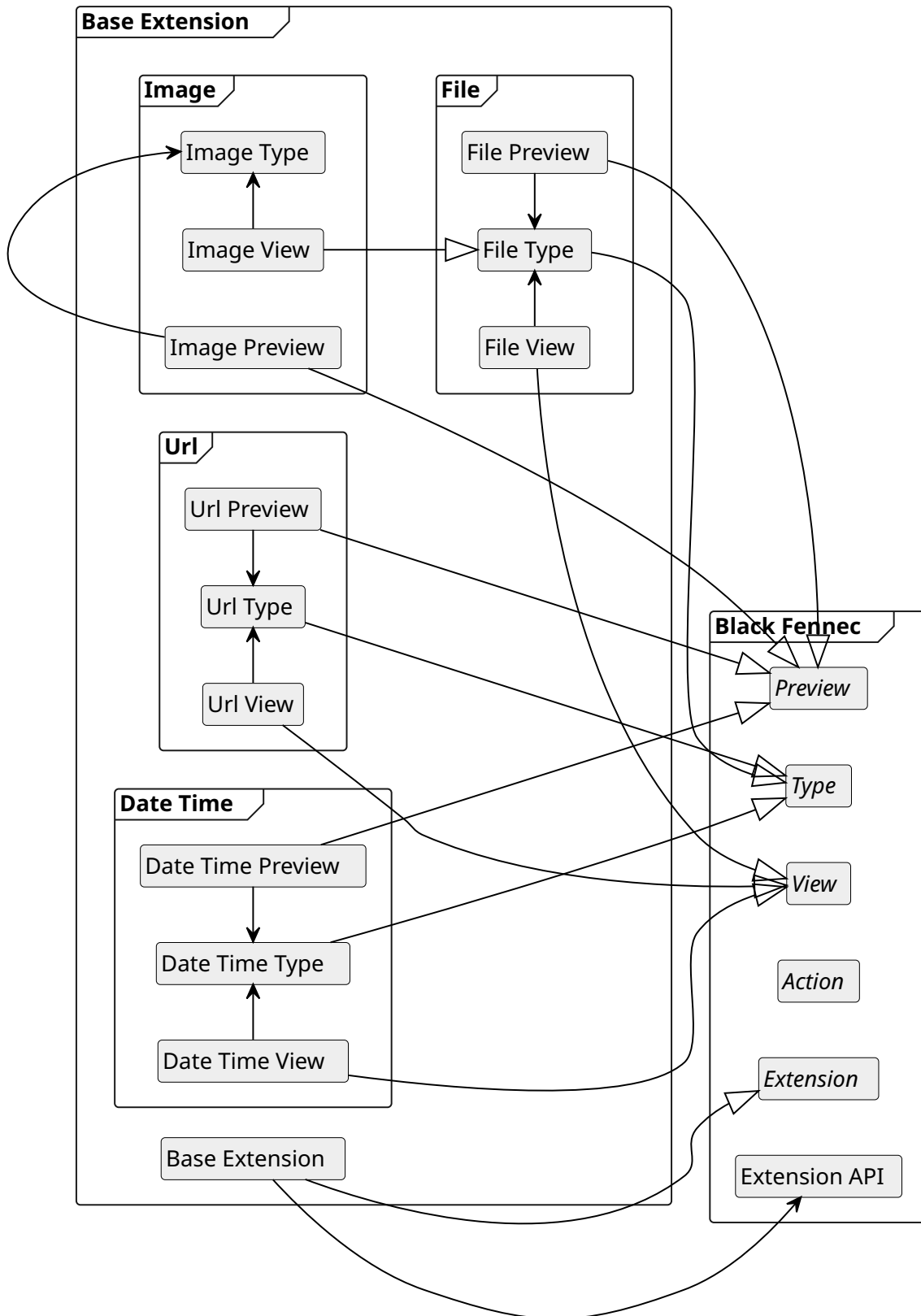
Core Extension



Base

The bae extension introduces novel types such as Image and Url as well as the associated *views*.

Base Extension

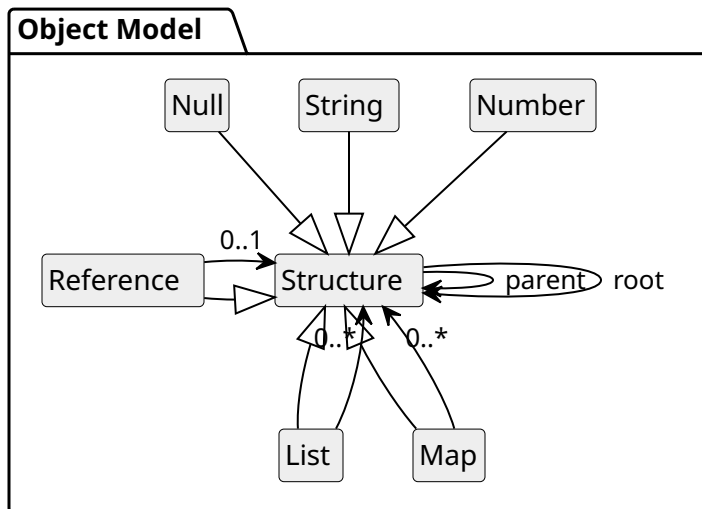


Structure

The object model (also known as core *types*) defines the fundamental types. All other types are structures built with it.

The object model is a DAG - apart from references - implemented with a composite pattern. It is possible to recognise this from the member of type structure in Lists and Maps. As for References, accessing the structure they point to is a special operation and References never has any children from the perspective of the DAG. Therefore, although they can break the DAG properties this is not their default behaviour.

In combination with the composite pattern the object model also includes a visitor pattern. This visitor pattern is for various utilities such as comparison, copying and also the *layering concept*.



Structure

Structure is the generic super class of all types in the object model. It is an abstract class and cannot be instantiated. Additionally to Structure there also exists a value structure which is the super class for all types that contain a value.

Parent of Structure

The parent of a structure is closer to the root in the *tree*.

Root of Structure

The root is a calculated property existing on each structure and is defined as a structure with no parent. In the *tree* it is the starting point of absolute paths.

Number

This core type represents numbers including integers and floating points.

String

This core type represents strings (e.g. “hello world”). It is also used for longer texts.

List

A list is a collection of structures.

Map

A map is a collection of key-value pairs of structures.

Reference

A reference is an absolute or relative path to structure. This type only exists on the *underlay* and is not visible on higher *layers*. The Reference consists of a list of Navigators.

Reference Navigation

A Navigator is a token of the parsed Reference. The list of Navigators contained in a *reference* are used to navigate through the *tree*.

Null

Null is a special type that represents the absence of a value.

Example

We will now look at an example of a structure. The following example is a simple structure given in the form of a JSON that represents an example person.

```
{
  "name": "max",
  "age": 50,
  "friends": [
    {"$ref": "#/wife"},
    "paul"
  ],
  "children": null,
  "wife": {
    "name": "eva",
    "age": 45
  }
}
```

(continues on next page)

(continued from previous page)

```
}  
}
```

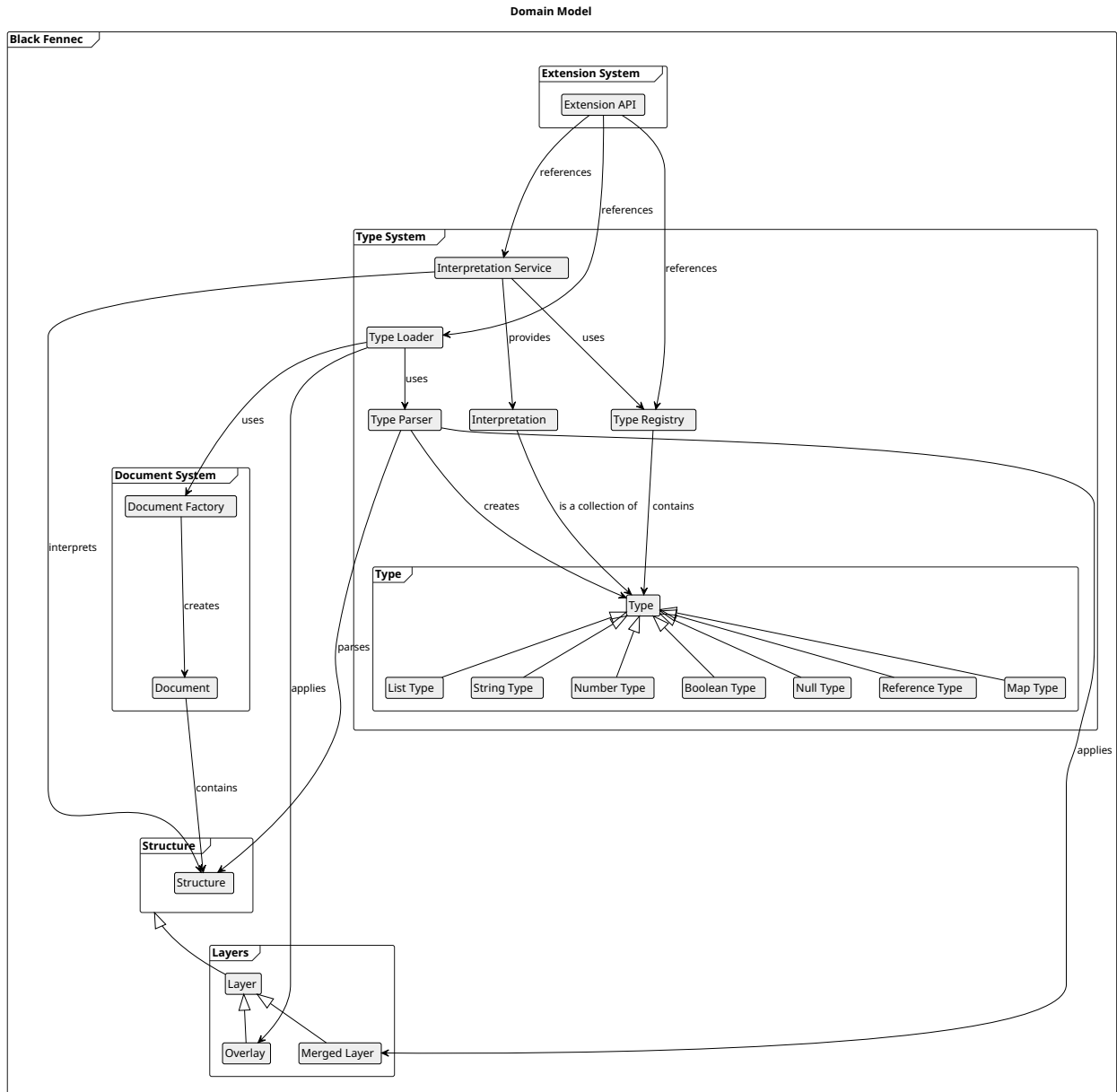
To represent this structure in the object model we would use the following code:

```
person = Map({  
  'name': String('max'),  
  'age': Number(50),  
  'friends': List([  
    Reference([RootNavigator(), ChildNavigator('wife')]),  
    String('paul')  
  ])  
  'children': Null()  
  'wife': Map({  
    'name': String('eva'),  
    'age': Number(45)  
  })  
})
```

Type System

The Type System in Black Fennec is responsible for defining and managing the various types that are used in the application. It consists of several key components, including the Type Registry, Type Parser, and Interpretation Service.

The Type System also includes various types themselves, such as Map Type, List Type, String Type, Number Type, Boolean Type, Null Type, and Reference Type. These types define the characteristics and behavior of the values they represent, and are used to ensure that values are stored and processed correctly within the application.

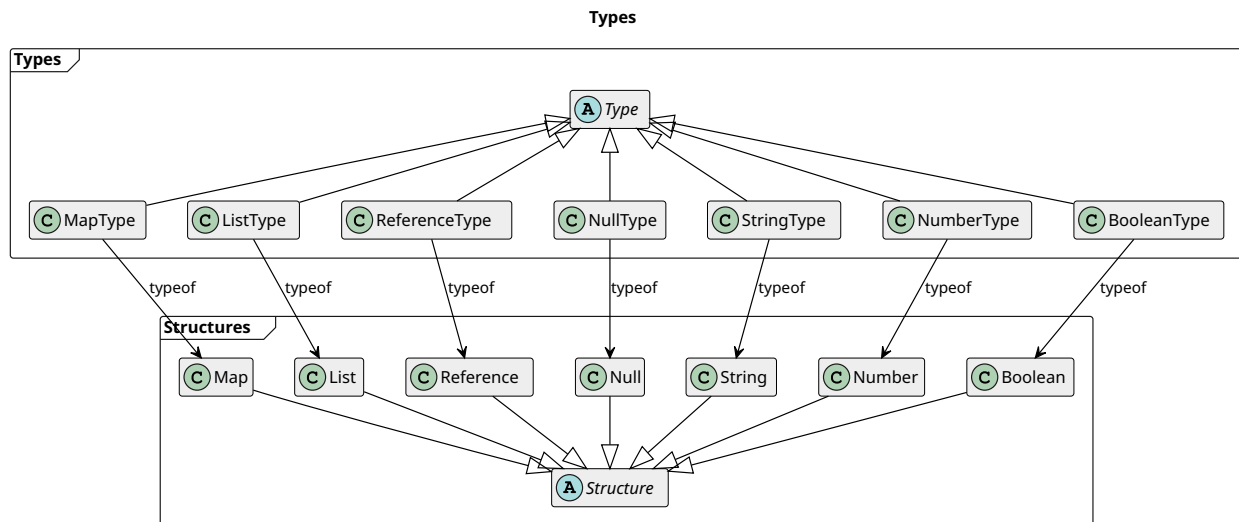


The Type Registry is a database of registered Types, which is used to store and manage the available types in the application. The Type Parser is responsible for parsing type definitions and creating new Types, while the Interpretation Service is responsible for interpreting values and mapping them to the appropriate Type.

Type

The concept of a Type does not differ significantly in Black Fennec when compared to a standard definition as used in many other programming languages. As such we will focus on the peculiarities of our implementation here.

The type system is based on core types which map one to one with core structures, as shown in the following diagram:



With the support from inheritance, the declarative syntax enables the reuse of definitions.

A type must inherit from *Type* and thus guarantees the existence of some properties like *create_instance*. Furthermore a type must be registered in the *Type Registry*. Otherwise it is unknown to the ecosystem, is not integrated, and thus the full potential of the type system remains unused.

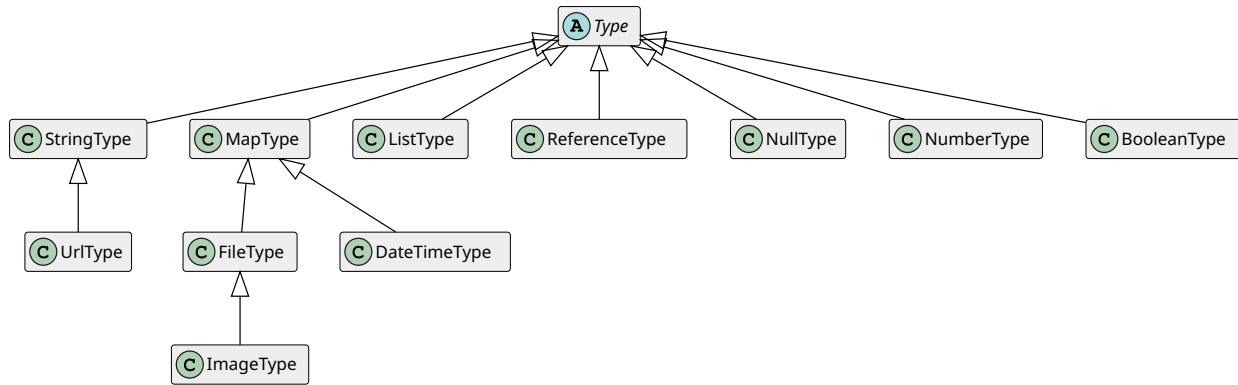
Types are the link between structures and most other components, including *Type View* and the *Action System*. As such they play a central role.

Types can be defined in different ways. The preferred way is by declaring it in a separate file and loading it with the *Type Loader*.

Inheritance

The Type System supports inheritance. This means that a type can inherit from another Type. The inheritance is transitive, meaning that a Type can inherit from a Type which inherits from another Type.

As such types are composites and subtypes of Core Types. This results in a type hierarchy, as shown in the following diagram:



Example: Image Type

If we wanted to improve the handling of images in Black Fennec we would probably want to add a *Type View* and allow *Actions* to be applied to them. The first step would be to define a *Type* so that images can be recognised and processed.

In this example we show how to define a type for images as seen in the json blow.

```

{
  "team_name": "Black Fennec",
  "team_logo": {
    "file_type": "image/png",
    "file_path": "path/to/image.png"
  }
}

```

Since images are files and we want to be able to handle them as such we will inherit from *File*. The file type can be defined as shown below.

```

{
  "super": { "$ref": "bftype://Map" },
  "type": "File",
  "required": [
    "file_path",
    "file_type"
  ],
  "properties": {
    "file_path": {
      "super": null,
      "type": "String"
    },
    "file_type": {
      "super": null,
      "type": "String"
    }
  }
}

```

The *super* field tells us that *File* inherits from *Map* and thus is able to define properties. The *type* field tells us that this is the type definition for *File*. The *required* field tells us that *file_path* and *file_type* are required properties. The *properties* field defines the properties of the type. In this case *file_path* and *file_type* are required to be of type *String*.

Now we can define the type for images. We will inherit from *File* and only override the *file_type* property with a *pattern* and a *default* value.

```
{
  "super": { "$ref": "bftype://File" },
  "type": "Image",
  "properties": {
    "file_type": {
      "pattern": "^image/.*$",
      "default": "image/unknown"
    }
  }
}
```

The *String* type allows us to define a *pattern* which allows us to define a regular expression that the value must match for a *Structure* to be recognised.

The *default* value is used to create instances of the type.

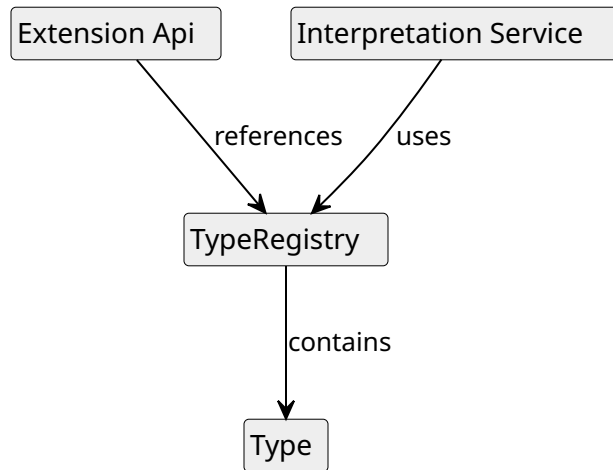
The *TypeLoader* will read the type definition from the file and register it in the *Type Registry*. In that process the type hierarchy is merged which will produce the following structure.

```
{
  "super": { },
  "type": "Image",
  "required": [
    "file_path",
    "file_type"
  ],
  "properties": {
    "file_path": {
      "super": null,
      "type": "String"
    },
    "file_type": {
      "super": null,
      "type": "String",
      "pattern": "^image/.*$",
      "default": "image/unknown"
    }
  }
}
```

After loading the type, the *Type Registry* will be able to recognize the type. Notice how the type definition and the structure are loosely coupled. It is indeed possible for a single structure to be considered valid for multiple types (e.g. *File* and *Image*). It is also possible that a structure matches a type but has additional attributes that are not part of the type definition. If you are interested in the interpretation of structures, checkout the *selection process*.

Type Registry

The Type Registry is a register of all known (aka registered) *Types*. Types which are not known to the type registry cannot be considered in the *selection process*. The type registry is accessible to extensions via *Extension API*.



Interpretation

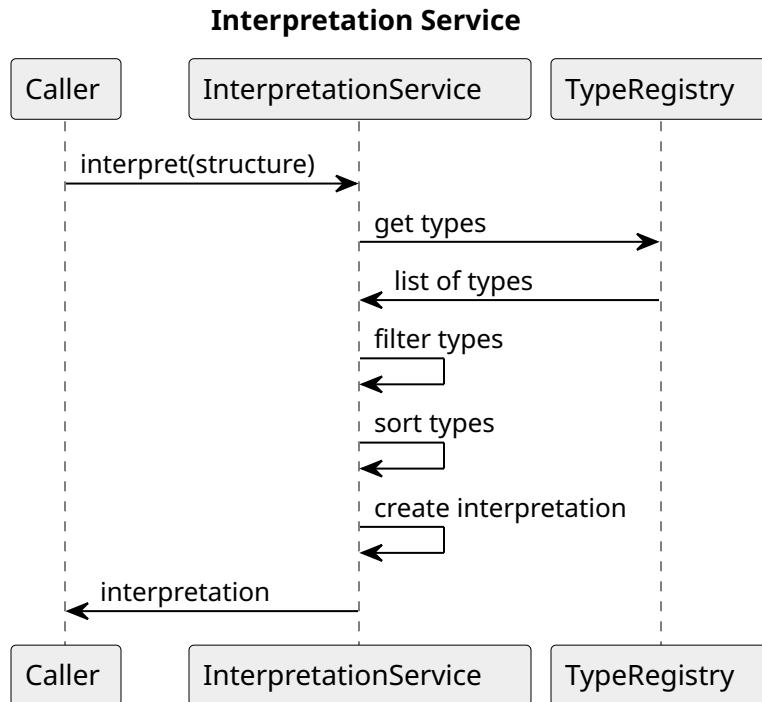
The process of interpretation solves the problem of composing what is considered the best interpretation of the underlying data. We sometimes refer to this process as an auction. Its exact implementation is subject to change and this document will cover the topic in general terms.

One of the main features of Black Fennec is to find the best interpretation of a given *Structure* using known *Types*. Types are considered known if they have been previously registered with the *Type Registry*.

Conceptually speaking - meaning that the following is not how we actually do it but how you can imagine it works - all registered types could be asked to produce a score (sometimes referred to as an offer) on how good they can represent the data structure. From the result of this process we decide which type will officially be the best interpretation of the data structure. The simplest possible implementation of this process would use a boolean as the score and select the first type that scores true.

Interpretation Service

The interpretation process is implemented in the Interpretation Service which in turn creates Interpretations. To create an Interpretation the Interpretation Service must evaluate and sort all known Types. The resulting sorted list of Types is then included in the Interpretation along with some metadata.



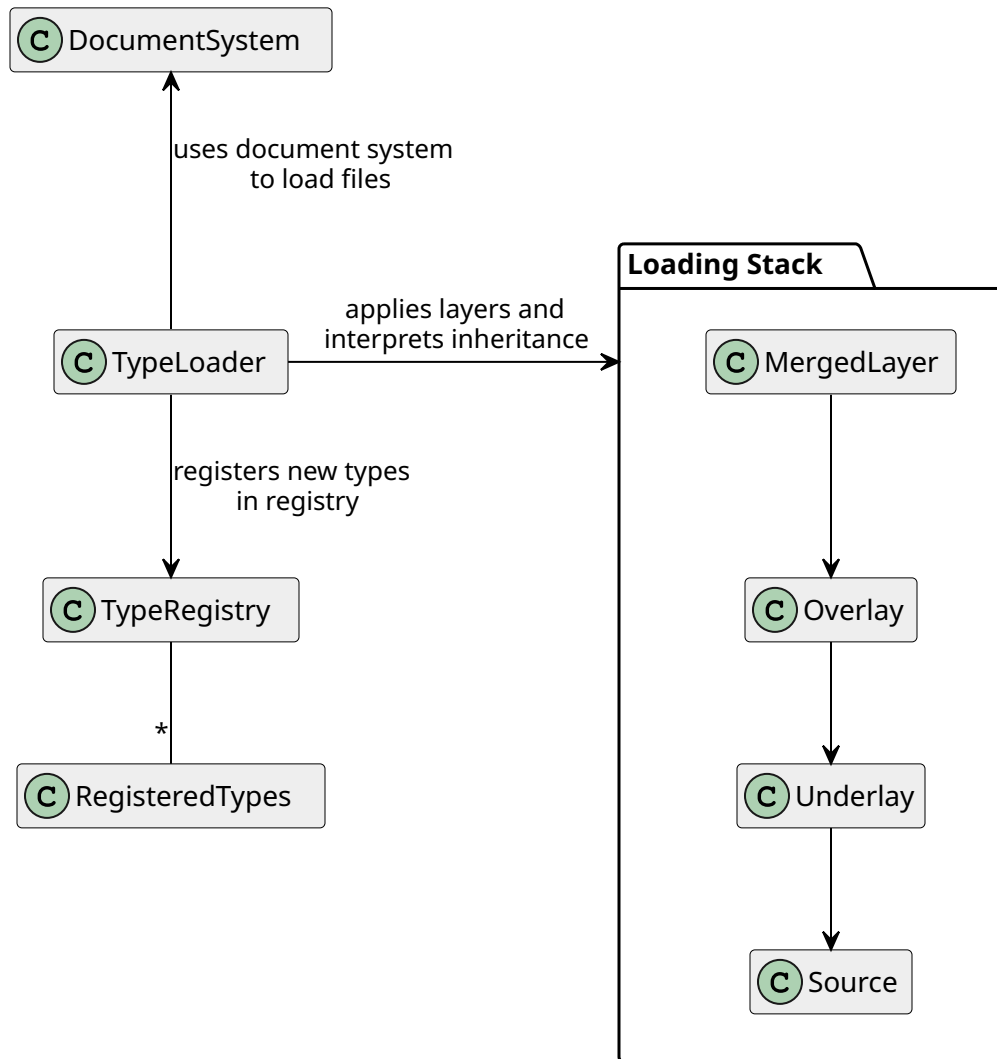
The caller of the Interpretation Service often uses the interpretation to later display the structure. Other use cases include the filtering of *Actions* based on the Interpretation.

Interpretation

The interpretation contains an ordered list of types associated with a structure. An interpretation created through the interpretation service is what Black Fennec believes to be the best available representation of a given structure.

Loading Types

Loading Types Overview



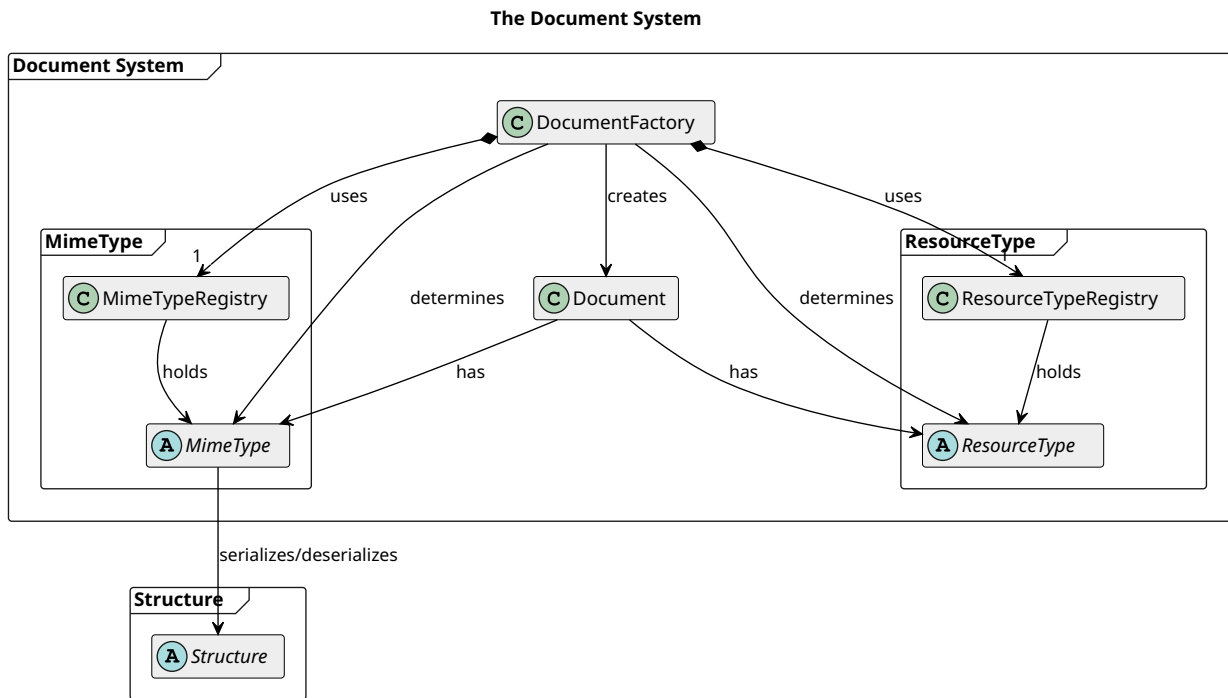
Type Loader

The Type Loader is able to load a Type from a file. It is available through the *Extension API* and can thus be used by extensions to define Types in a declarative way. Internally, the Type Loader uses the *Document System* to load the files, allowing the extensions to define types in any supported mime type and from any supported resource. Further it applies an *Overlay* such that references to other types are resolved correctly. This is used in combination with the *Black Fennec Type Resource Type* to implement inheritance.

Document System

The document system is a set of strategies that allows Black Fenec to be completely independent of the underlying file types. It is a set of interfaces that allow Black Fenec to read and write files in a generic way. The document system is also responsible for the file type detection and the file type association.

The overall interaction of the different components involved is visualized in the following diagram:



Document

A document is a file that can be opened by Black Fenec. It is a set of data that can be read and written by Black Fenec. A document is traditionally a representation of a file. It leverages the *Resource Type* and *Mime Type* strategy patterns to load and parse arbitrary file types from arbitrary sources. A document can easily be created via the *Document Factory*.

Document Factory

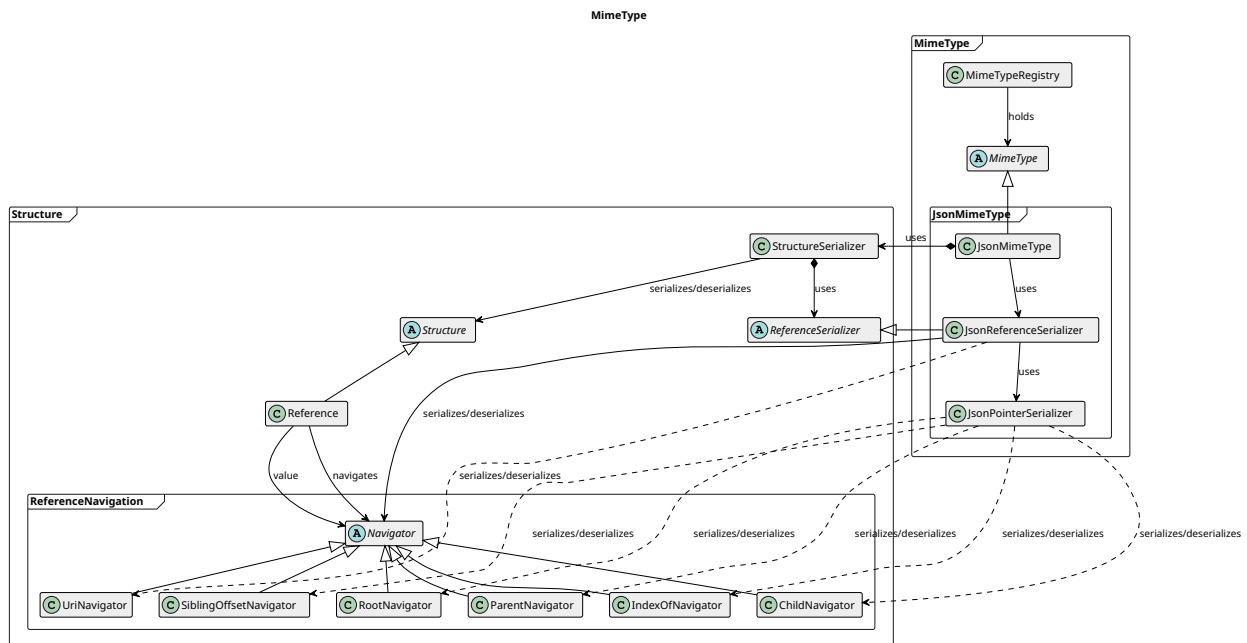
The Document Factory is a factory that creates *Document* instances. It can be responsible for the resource/mime type detection and does the resource/mime type lookup in the *Resource Type Registry/Mime Type Registry*. These Registries are to be dependency injected upon creation of the Document Factory.

Mime Type

Mime Types handle the different content types that are supported by the application. The Mime Type is an abstract concept, that has an implementation for each concrete Mime Type.

The Abstract Mime Type also has the responsibility to determine the mime type of an arbitrary URI. It uses different approaches to achieve this, such as path endings, or mime type of http responses for the given *Resource Type*.

As example of how different components interact, the following diagram contains the concrete implementation of the JSON Mime Type which intends to make the interactions more clear.



The diagram also contains the Structure Package with the Reference as a concrete implementation of the abstract class Structure. While not shown in the Diagram all Structures such as Map, List, String, etc. can be serialized and deserialized as it is required by the Liskov Principle. The Reference was chosen as an example because it is the most complex Structure that additionally contains Navigators which have to be handled specifically by the Mime Type.

Mime Type Registry

The Mime Type Registry contains all registered Mime Types and can be used as a lookup service. Mime Types typically get registered at the start of the application and get deregistered at the end. The registry allows for Extensions to register their Mime Types.

Json Mime Type

The Json Mime Type is the first MimeType implemented. It is used to serialize and deserialize the data to and from JSON.

Json Reference Serializer

The Reference Serializer is required by the Structure Serializer as it contains the strategy of how references for the specific mime type (here JSON) can be serialized and deserialized. Json References are specified in the form of URIs. Black Fennec can handle any *Resource Type* and *Mime Type* that is contained in the respective registries.

Json Reference Pointer

The JSON mime type also provides a Json Pointer Serializer which is able to serialize the part of the JSON Reference contained in the fragment. This JSON Pointer is implemented according to the IETF Specification (RFC 6901).

InMemory Mime Type

The InMemory mime type is used to reference already loaded types. It is essentially a mock mime type that does not serialize or deserialize anything. Currently it is impossible to save structures loaded with this mime type.

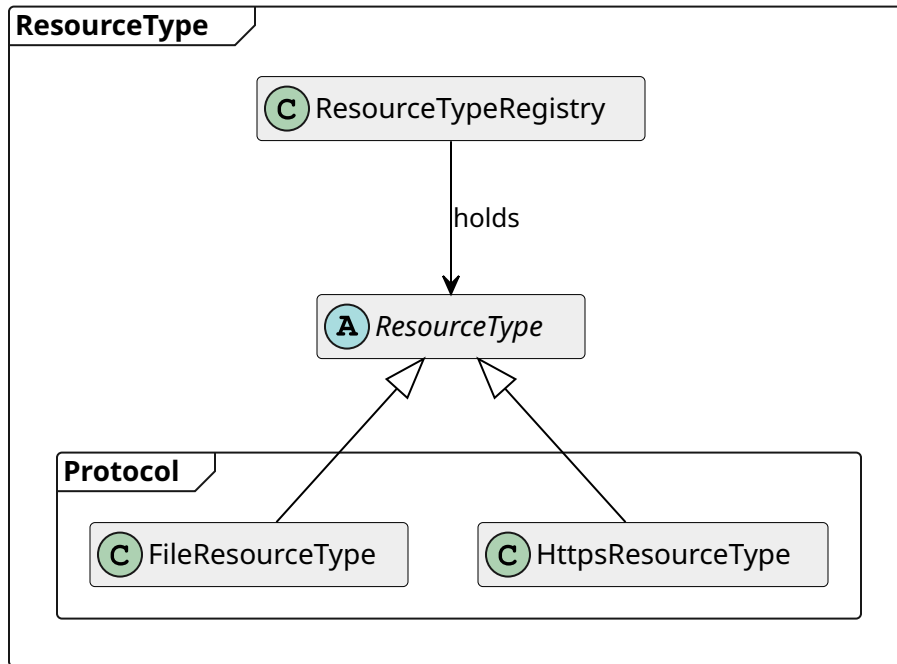
Resource Type

Resource Types handle the different data resources that are supported by the application. The Resource Type is an abstract concept, that has an implementation for each concrete Resource Type. As example of how different components interact, the following diagram contains the concrete implementation of the Http Resource Type which makes the interactions more clear.

The Abstract Resource Type has the responsibility to determine the right resource type from an arbitrary URI. This is done via the Scheme/Protocol contained in the given URI. If not protocol can be found the fallback type 'file' is used.

As example of how different components interact, the following diagram contains the concrete implementation of the Http Resource Type which intends to make the interactions more clear.

ResourceType



Resource Type Registry

The Resource Type Registry contains all registered Resource Types and can be used as a lookup service. Resource Types typically get registered at the start of the application and get deregistered at the end. The registry allows for Extensions to register their own Resource Types.

Https Resource Type

The Https Resource Type is a concrete implementation of the Abstract Resource Type. It is responsible for handling all Https requests. This allows Black Fennec to load data from remote servers in the same form as it would handle a local file.

File Resource Type

The File Resource Type is a concrete implementation of the Abstract Resource Type. It is responsible for handling all file requests. This allows Black Fennec to load data from local files.

BfType Resource Type

The BfType Resource Type is a concrete implementation of the Abstract Resource Type. It is responsible for handling the *bftype://* protocol. This allows Black Fennec to reference *types* which are currently loaded in the *Type Registry*.

Action System

The action system is a set of functions that are called when a certain event occurs. The event can be a user action, such as clicking a button, or a system event, such as a timer expiring. The action system is used allow automated modification of the structure.

Action

An action is a function that can be executed by the user from the GUI and are able to manipulate the structure.

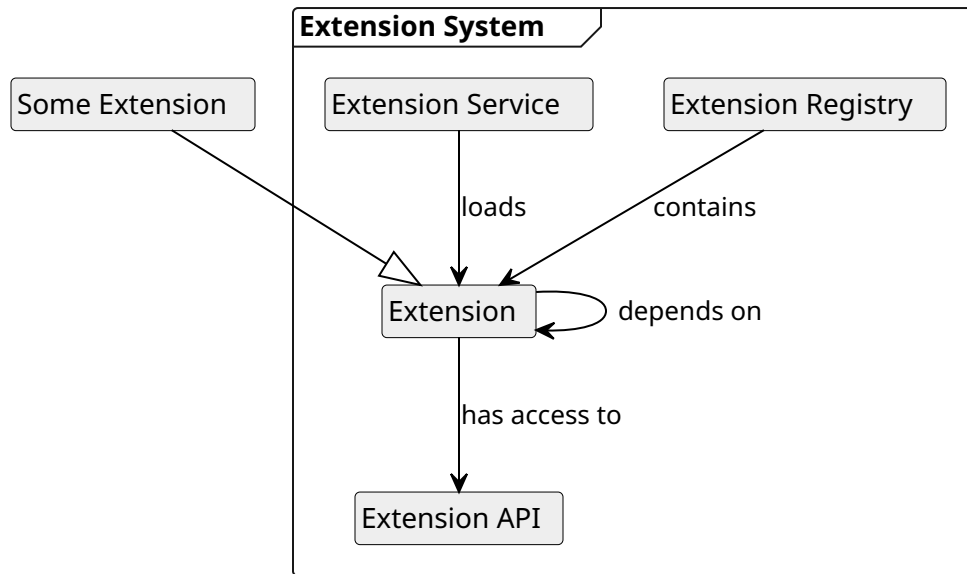
Action Registry

The action registry is a dictionary that maps action names to action functions. The action registry is used to register actions thus make them available to other parts of the application.

Extension System

Black Fennec application’s capabilities and usability rely heavily on the availability of extensions. The Extension System is designed to be stable, with the goal of maintaining backward compatibility throughout the 1.x releases cycle. The Extension System is responsible for loading extensions, managing their lifecycle, and providing the extension API to extensions. The extension registry stores and organizes the extensions, and is used in part for dependency resolution. The following diagram illustrates the conceptual landscape of the extension system:

Conceptual Extension Landscape



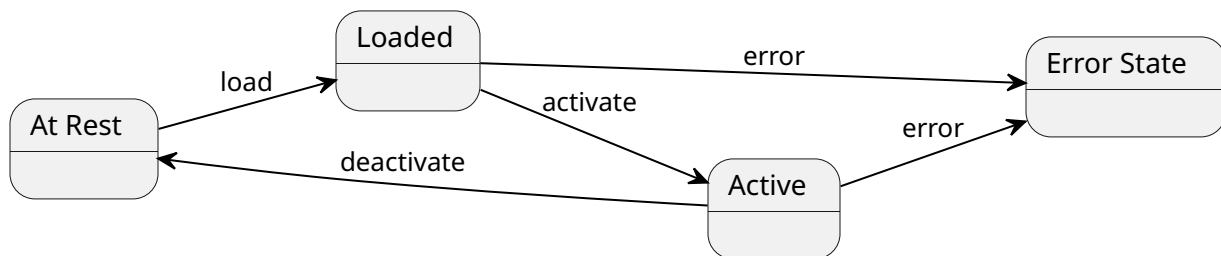
The components of the extension system, including the Extension, Extension Registry, Extension API, and Extension Service, are described in more detail on the linked pages.

Extension

Extensions allow developers to extend and customize the application with various components and functionality. Extensions can provide various types of components, including types, type views, presenters, actions, and more. Extensions interact with the system via the *Extension Api*.

The *Extension System* in Black Fennec manages the lifecycle of extensions, which includes four states: at rest, loaded, active, and error state.

Extension Lifecycle



When an extension is first installed, it is in the “at rest” state. When the Extension Service loads the extension, it

transitions to the “loaded” state. From there, the extension can be activated and move to the “active” state, allowing it to utilize the features and functionality of the Black Fennec application. If an error occurs while the extension is active, it will transition to the “error state”. In this state the extension can be reset and return to the “at rest” state.

Extensions can also define dependencies on other extensions, which means that they require certain other extensions to be installed and loaded in order to function properly. This helps to ensure that the necessary components and functionality are available to the extension.

There are also some recommended extensions that are commonly used in Black Fennec installations. These include the Black Fennec Core extension, which provides the basic type views, a presenter, and some actions, and the Black Fennec Base extension, which provides commonly used types such as dates, URLs, and images.

```
class blackfennec.extension_system.extension.Extension(name: str, api: ExtensionApi, dependencies:
                                                    Optional[set[str]] = None)
```

Bases: object

```
class State(value, names=None, *, module=None, qualname=None, type=None, start=1,
            boundary=None)
```

Bases: Enum

```
ACTIVE = 'active'
```

```
DEPENDENCY_MISSING = 'dependency_missing'
```

```
FAILED = 'failed'
```

```
INACTIVE = 'inactive'
```

```
activate() → None
```

```
deactivate() → None
```

```
property dependencies: set[str]
```

```
deregister_actions()
```

```
deregister_presenters()
```

```
deregister_types()
```

```
deregister_view_factories()
```

```
property is_active: bool
```

```
property name: str
```

```
register_actions()
```

```
register_presenters()
```

```
register_types()
```

```
register_view_factories()
```

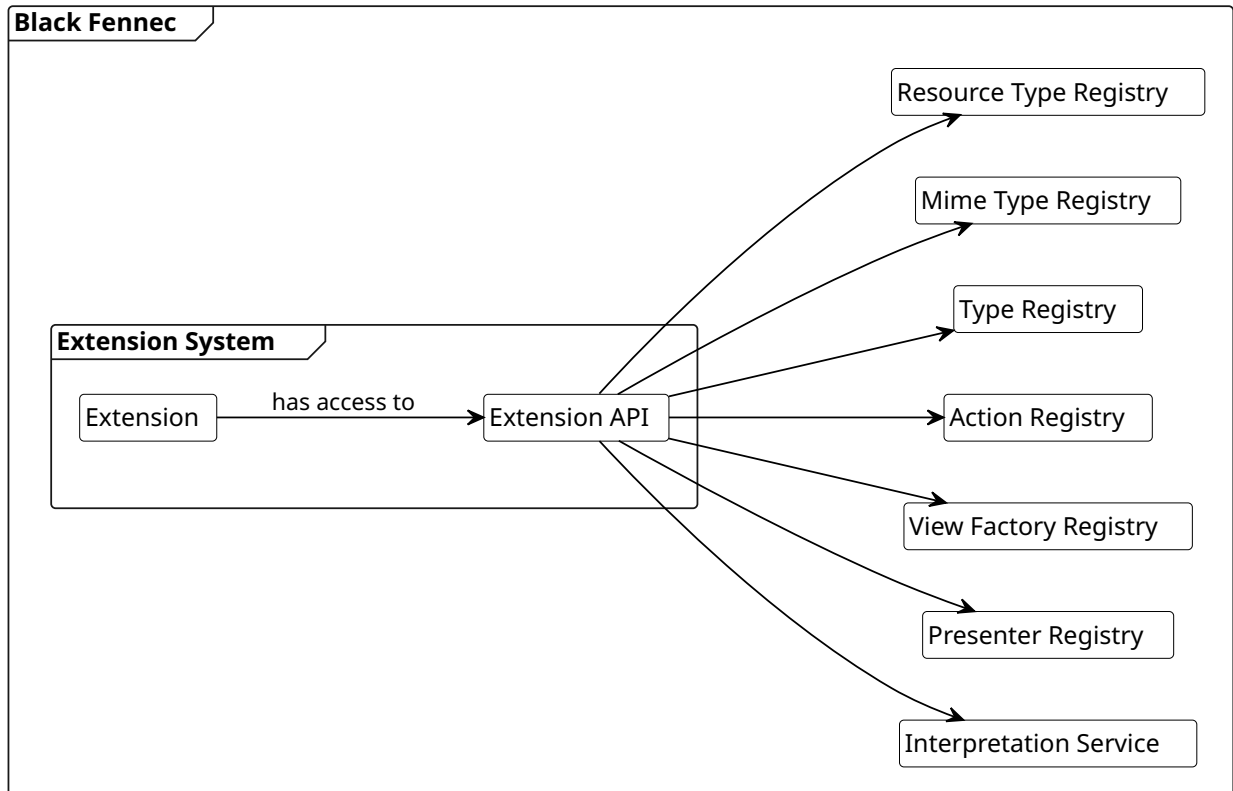
```
property state: State
```

Extension Api

The Extension API providing a way for *extensions* to access various features and functionality within the application. It acts as an interface between extensions and the application, allowing extensions to interact with and utilize the available services and resources.

The Extension API consists of a collection of services. These services include, the *Interpretation Service*, the *Presenter Registry*, the *View Factory Registry*, the *Action Registry*, the *Type Registry*, the *Mime Type Registry*, and the *Resource Type Registry*.

Extension API



The generated code documentation below provides a up-to-date description of the interface the Extension Api provides to extensions.

```

class blackfennec.extension_system.extension_api.ExtensionApi(presenter_registry:
    PresenterRegistry, type_registry:
    TypeRegistry,
    interpretation_service:
    InterpretationService, view_factory:
    view_factory_registry, type_loader,
    action_registry: ActionRegistry,
    document_registry:
    DocumentRegistry,
    document_factory:
    DocumentFactory, ui_service:
    UiService, mime_type_registry:
    MimeTypeRegistry,
    resource_type_registry:
    ResourceTypeRegistry)

```

Bases: object

Enables extensions to load types, actions or presenters with services and registries contained within the this class.

```

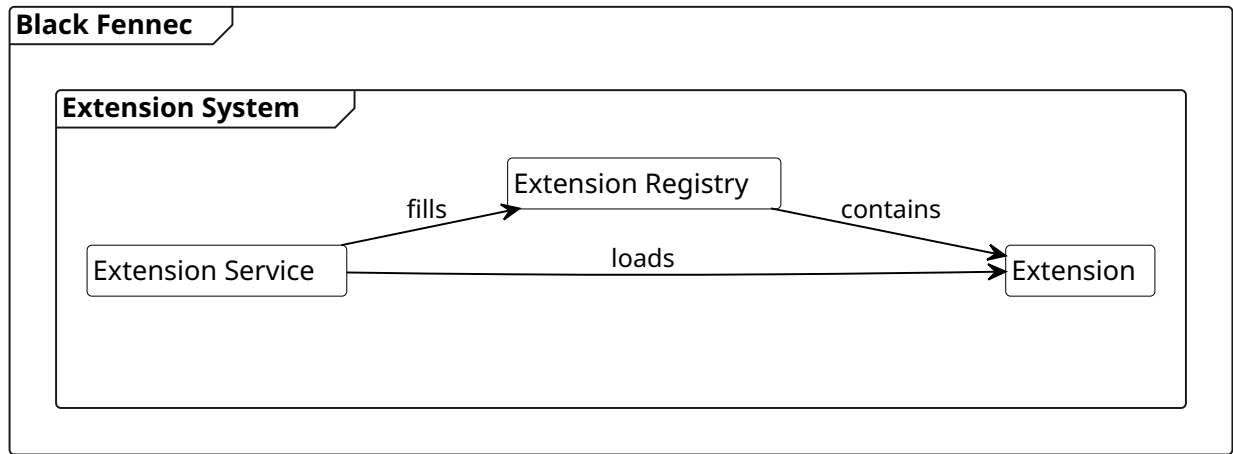
property action_registry: ActionRegistry
property document_factory: DocumentFactory
property document_registry: DocumentRegistry
property interpretation_service
property mime_type_registry: MimeTypeRegistry
property presenter_registry: PresenterRegistry
property resource_type_registry: ResourceTypeRegistry
property type_loader
property type_registry: TypeRegistry
property ui_service: UiService
property view_factory
property view_factory_registry

```

Extension Service

The Extension Service is an important component of the Extension System responsible for loading and initializing extensions, and ensuring that they are properly integrated into the application.

Extension Service

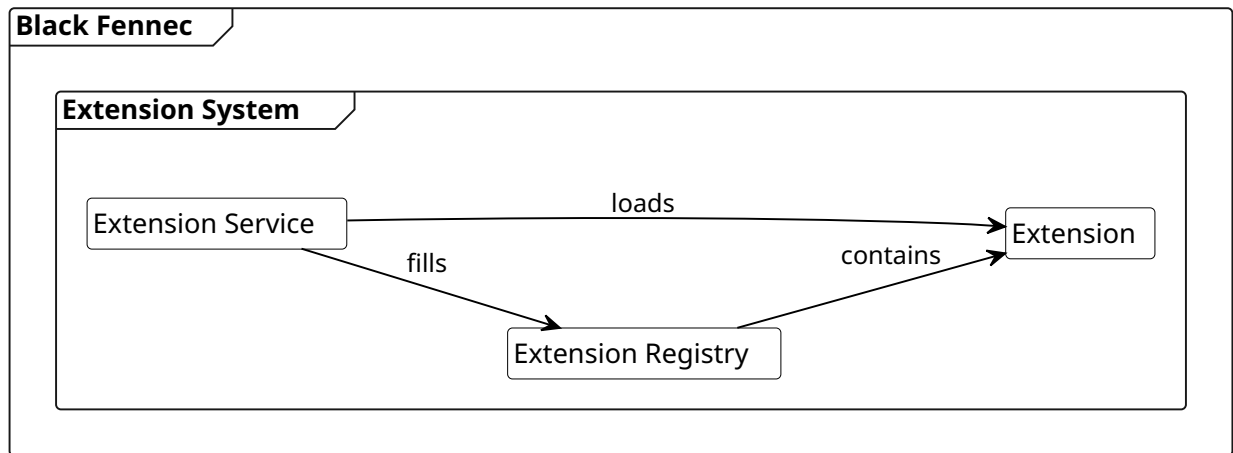


In order to load extensions, the Extension Service uses Python entry points to search for and locate extensions within the environment. It then resolves the dependency tree and defines the order in which the extensions should be loaded. Once the extensions are loaded, the Extension Service registers them in the *Extension Registry*, making them available to be accessed and utilized by other parts of the application. The Extension Service plays a crucial role in ensuring that the Extension System is functioning properly and that extensions are able to interact with and utilize the features and functionality of the Black Fennec application.

Extension Registry

The Extension Registry is a component of the *Extension System* in the Black Fennec application. It stores the available *extensions*, allowing them to be accessed and utilized by other parts of the application.

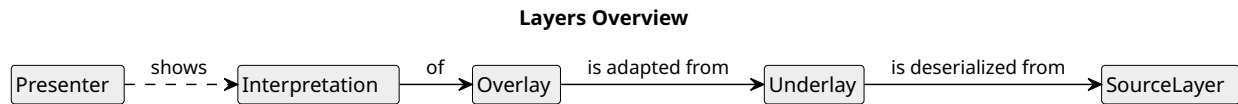
Extension Registry



The Extension Registry is responsible for maintaining a list of all the available extensions within the application. It allows the Extension Service to load and initialize extensions, and provides a way for other parts of the application to access and utilize extensions. The Extension Registry also plays a role in dependency resolution, providing the data to ensure that extensions are loaded in the correct order.

Layers

The different levels of abstraction in this model can be thought of as layers. Each layer adds (or removes) something to the layer below it. Services typically exist in a single layer, as they are not typically aware of the concept of layers. The layers above the underlay should be transparent, which allows for arbitrary combinations to be adapted to different use cases.



Presenter

The *Presenter* is responsible for presenting the data to the user in a meaningful way. It uses the *Interpretation* and uses it to create *Views* which it display to the user. This may involve adding additional context or information, or interacting with the user in some way.

Interpretation

The *Interpretation* represents the data in a form that is easier for the Presenter to understand. It is created by the *Interpretation Service*, which processes the data and returns matching *Types*

Overlay

The *Overlay* is the processed *Underlay*. It is responsible for resolving references within the data, which allows the data to be connected and linked together in a meaningful way. This may involve resolving references to external sources, or linking related pieces of data together.

Underlay

The *Underlay* is the deserialized *Source Layer* in the form of a structure. It represents the raw data in a more organized form, which makes it easier for the Overlay to process and resolve references. The Underlay is the starting point for the data processing pipeline, and all subsequent layers build on top of it.

Source Layer

The *Source Layer* is the source of the data. It can be thought of as the raw, unprocessed data that is fed into the data processing pipeline. Currently, only JSON is officially supported as a source, but it is possible to extend the system to support other mime types such as XML or YAML.

These layers provide a high-level overview of the data processing pipeline, but there are many other layers at play as well. For more information on all layers see the table of contents below.

Observable Layer

The Observable Layer is a feature that allows you to monitor changes to a structure. It does not modify the structure in any way, but instead adds observability to changes made to the structure by other entities. This means that any observer registered with the Observable Layer will be notified every time a change is made to the structure.

One practical application of the Observable Layer is in the implementation of an “undo” and “redo” function. The *History Service*, for example, can use the observable layer to track changes made to a structure and allow users to undo or redo those changes as needed.

Here’s an example of how the Observable Layer might be used:

```
# same as above but in Python using Black Fennec API
structure = Map({
  "key": String("initial value")
})
```

(continues on next page)

(continued from previous page)

```

layer = ObservableLayer()
observed_structure = layer.apply(structure)

def on_change(sender, notification):
    print(f"Structure has been updated to: {notification.new_value}")

observed_structure.bind(changed=on_change)

structure["key"] = String("new value")

# Output: "Structure has been updated to: { key: 'new value' }"

```

In this example, we create a new `ObservableLayer` using an existing structure as the base. We then subscribe to changes to the structure using the `bind()` method. When we update the structure by changing the value of the key field, the observable layer notifies our observer, causing the callback function to be executed.

Overall, the `Observable Layer` is a powerful tool for tracking changes to a structure and reacting to those changes in real-time.

Merged Layer

The `Merged Layer` is a feature that allows you to combine two *Structures* into a single structure. It is different from regular layers in that it requires two input structures instead of one, and the output is a completely new structure that cannot be traced back to its original inputs. This is demonstrated in the following example:

Consider the following two structures:

```

{
  "example": {
    "title": "Merged Layer Example",
    "text": "In this example we will merge two structures."
  },
  "merged": {
    "rating": 5,
    "comments": [
      "This is a comment.",
      "This is another comment."
    ]
  }
}

```

If we merge the example and merged objects, the resulting structure would be:

```

{
  "title": "Merged Layer Example",
  "text": "In this example we will merge two structures.",
  "rating": 5,
  "comments": [
    "This is a comment.",
    "This is another comment."
  ]
}

```

It's worth noting that the Merged Layer is currently not editable, meaning that it cannot be modified after it has been created. Despite this limitation, the merged layer plays a crucial role in implementing *inheritance* within the *Type System* allowing the definition of a type to be merged with the definition of its super type, similar to the `__proto__` property in JavaScript.

Shadowing

Shadowing is a phenomenon that occurs when both structures to be merged define the same key. In this case, the value in the overlaying structure “shadows” or hides the value of the underlying structure. This means that the value from the overlaying structure will be used in the merged structure, while the value from the underlying structure will not be visible.

For example, consider the following two structures:

```
{
  "underlay": {
    "key": "underlay value"
  },
  "overlay": {
    "key": "overlay value"
  }
}
```

If we merge the underlay and overlay objects, the resulting structure would be:

```
{
  "key": "overlay value"
}
```

In this case, the value of the key field in the underlying structure is “shadowed” or hidden by the value of the same field in the overlaying structure. As a result, the value from the overlaying structure is used in the merged structure, while the value from the underlying structure is discarded at this layer.

It's worth noting that shadowing can occur at any level of the structure, not just at the top level. For example, if the underlay and overlay structures were both nested objects, shadowing could occur at any level of the nested objects.

Overlay

The responsibility of the Overlay is the resolutions of *References*. The resulting layer thus replaces all references with the referenced object. This is best illustrated with an example.

A reference in JSON could look like this:

```
{
  "example": {
    "title": "Reference Resolution Example",
    "text": "This is an example of a reference resolution."
  },
  "reference": { "$ref": "#/example" }
}
```

After applying the Overlay, the reference is resolved and the resulting *Structure* feels like this:

```
{
  "example": {
    "title": "Reference Resolution Example",
    "text": "This is an example of a reference resolution."
  },
  "reference": {
    "title": "Reference Resolution Example",
    "text": "This is an example of a reference resolution."
  }
}
```

Note, that it only *feels* like this. In actuality there is no duplication of the object. The reference is resolved and replaced with a Python reference, thus changes are always visible in both places.

Since this feature is implemented in a *Layer* it is non-destructive and the original data is not modified. This allows us to “recover” all used JSON references when saving the data.

Underlay

Underlay is a synonym for *Structure* and is the deserialized data from the *source*. As such it is the content of a *Document* as created by the *Document Factory*.

This virtual layer is closest to the *source* and has not been modified by any *layer*. However, it is possible that the *Document System* has modified the *source* in order to create the *underlay*. These modifications are not visible to Black Fennec and will most likely be lost when the data is saved.

Source Layer

The source layer represents the actual source of the data and is not a concept found in any other part of the documentation. It can be thought of as the raw, unprocessed data that is fed into the system. Currently, only JSON is officially supported as a source, but it is possible to extend the system to support other mime types such as XML or YAML. The actual implementation of the source layer is realized with the Document System, which is responsible for handling the deserialization and organization of the raw data.

Presentation System

The presentation system is a key component of the Black Fennec software, responsible for the visual representation of information within the software and facilitating user interaction with the various components of the software. It includes a range of features and components, such as the *UI Service*, *Navigation Service*, and *History Service*, that enable the software to effectively display and interact with data.

The *Type View* class is a base class for all views for structures in the Black Fennec software. A view is a widget that represents the visual representation of a type within the software, serving as a means of displaying information to the user and facilitating interaction with the software.

The *Presenter* class is responsible for rendering the content of a tab within the software, displaying the complete contents of a file and serving as a container for other components that define how these components are bound together. One specific implementation of the Presenter is the column-based presenter, which is able to show a single level of a structure in a single column and respond to navigation requests received from the structure rendered in that column.

The Navigation Service is responsible for facilitating navigation between different views and structures within the software, forwarding presentation requests in the form of structure requests to the presenter and enabling a more intuitive and seamless user experience when navigating between different views.

The *UI Service* serves as the interface between the various extensions and the rest of the code base, providing access to a range of predefined UI components that can be utilized to enhance the user experience and facilitate the execution of various tasks within the software. It also acts as an interface to the clipboard and toast messages, allowing the extensions to interact with these features in a seamless and intuitive manner.

The *History Service* is responsible for maintaining a record of the user's interactions with the software, enabling the user to navigate backwards and forwards through their history of actions within the software.

In addition to the various components described above, the presentation system also includes all views and view models of the application. But these are not described in detail here, as they are not part of the domain model.

Together, these components of the presentation system work together to decouple the rest of the code base from the UI, allowing the software to be independent of the UI to some extent.

Following is a table of contents of the aforementioned components of the presentation system:

Type View

The view is a base class for all views for structures in the Black Fennec software. A view is a widget that represents the visual representation of a type within the software. It serves as a means of displaying information to the user and facilitating interaction with the software.

Together, the type views and type view factories play important roles in the presentation and visualization of information within the Black Fennec software, enabling the software to display and interact with data in a flexible and extensible manner. These two classes are primarily used by the Extensions which define custom views for specific *types*.

Type View Factory

The type view factory class is a base class for all view factories in the Black Fennec software. A view factory is a class that creates views for a specific type. It has two main methods: *satisfies* and *create*. The *satisfies* method tests if the view factory can satisfy a given specification, returning a boolean value indicating whether the specification can be satisfied. This can then be leveraged by a presenter or also another view responsible of showing child views to specify what kind of view should be created. The *create* method creates a view based on a given interpretation, which is an overarching representation of an interpreted structure.

Type View Factory Registry

The Type view factory registry is a central registry for all type view factories in the Black Fennec software. It serves as a point of connection between extensions that provide view factories and the various components of the software that require them. Extension developers can register their view factories with the registry, making them available for use by the software, while the software can retrieve type view factories from the registry as needed.

The type view factories contained are registered bound together with their type and also a specification that defines the use and capabilities of the type view.

Structure View Factory

The structure view factory is a more general view factory that is not specialized on a specific *type* and creates views for all *interpretations* of a structure. Thus it has the responsibility to iterate over an *interpretation* and create type views with the help of the type view factory for all interpreted *types*. It is used by a *Presenter* to create a view for a *structure* `<definition_structure>`.

Presenter

A presenter is a component responsible for rendering the content of a tab. This means that it has the ability to display the complete contents of a file. However, it is not necessarily required to render the structures contained within the file, but rather acts as a container for other components that define how these components are bound together.

One specific implementation of the presenter is the column-based presenter, which is able to show a single level of a structure in a single column and respond to navigation requests received from the structure rendered in that column. When a navigational request is received, the column-based presenter is responsible for controlling how the navigation impacts the creation or closing of columns in the presentation.

Presenter Registry

The presenter registry is a central repository for all presenters within the Black Fennec software. It serves as a point of connection between extensions that provide presenters and the application that requires a presenter when opening a new tab. Extension developers can register their presenters with the registry, making them available for use by the application, while the application can retrieve presenters from the registry as needed.

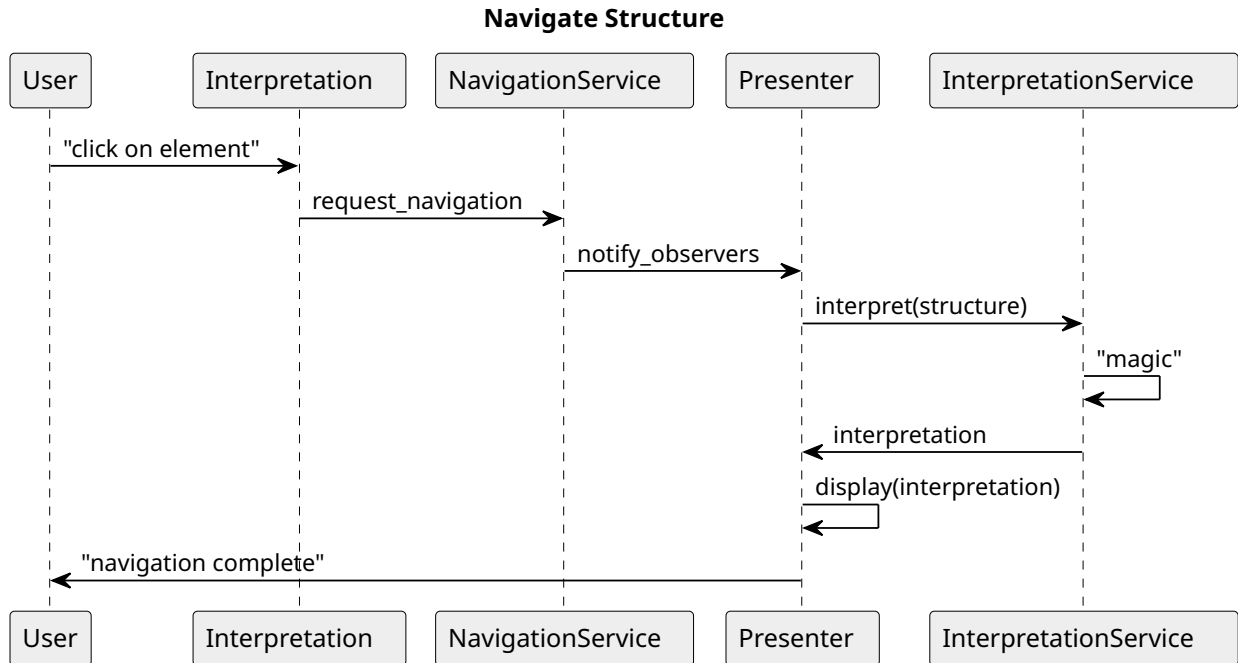
Navigation Service

The NavigationService provides the means to request the navigation to a structure. The NavigationService forwards the request in the form of a presentation request to the *Presenter*. It is the responsibility of the presenter to get an appropriate interpretation of the structure.

One key aspect of the Navigation Service is its ability to support navigation between different views. In a nutshell a *view* is a visual representation of a *structure* within the software.

Navigation

A *view* might not want to display the entire structure it represents by itself. For example the *List* does not render its content in full. Instead the view provides a clickable area (e.g a button) which when pressed should take the user to the substructure. This process is called navigation.



The navigation itself in the end is handled by the *presenter*. The Navigation Service is only responsible for forwarding the request to the presenter.

UI Service

The UI Service serves as the interface between the various extensions and predefined UI utilities and components. These components can be utilized by the extensions to enhance the user experience and facilitate the execution of various tasks within the software.

In addition to providing access to UI components, the UI Service also acts as an interface to the clipboard and toast messages, allowing the extensions to interact with these features in a seamless and intuitive manner. This enables the extensions to perform a wide range of functions, such as copying and pasting data, displaying notifications to the user, and more.

The UI Service in the Black Fennec software maintains a map of multiple contexts, each representing a distinct instance of an application window. This feature is useful for ensuring that UI elements such as dialogs are properly associated with the correct parent window. For example, when a function call is made to the UI Service to instantiate a dialog, a context can be passed along with the call to specify which window should serve as the parent. This allows the instantiator to define the transiency of the dialog relative to the parent window, ensuring a more intuitive and seamless user experience.

Overall, the ability to specify a context in function calls to the UI Service is an important aspect of its functionality, as it enables the software to properly handle the relationships between different UI elements and application windows. This helps to enhance the usability of the Black Fennec software and facilitates the execution of various tasks within the software.

History Service

The History Service is a system that allows users to undo and redo actions within the Structure. It works by tracking changes made to a data structure and storing a history of those changes. The History Service typically makes use of the *Observable Layer*, which allows it to monitor changes to an entire *Structure* and store a history of those changes. When the user requests an “undo” or “redo” action, the History Service uses this history to revert the structure to a previous state or move forward to a later state.

Here’s an example of how the History Service might be used to offer “undo” and “redo” functionality in a UI:

```
structure = Map({
    "key": String("initial value")
})

layer = ObservableLayer()
observable_structure = layer.apply(structure)

history_service = HistoryService(layer)

# Register a change to the structure
structure["key"] = String("new value")

# Undo the last change
history_service.undo()

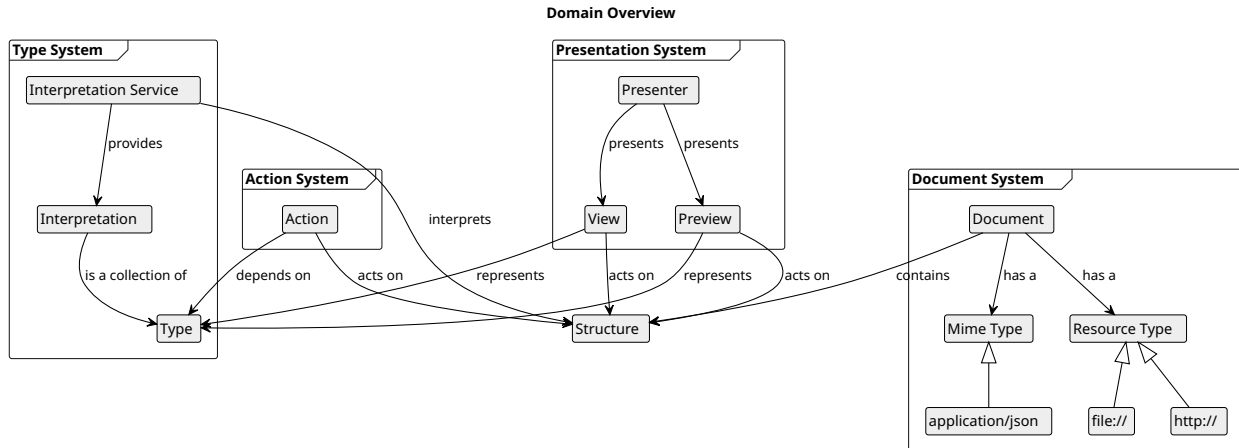
print(structure["key"])
# Output: "initial value"

# Redo the last change
history_service.redo()

print(structure["key"])
# Output: "new value"
```

In this example, we create a new Observable Layer using an existing Structure as the base, and we create a new History Service using the Observable Layer as the source of change history. We then register a change to the Structure by updating the value of the key field. When we call the `undo()` method on the History Service, it reverts the Structure to its previous state, changing the value of the key field back to “initial value”. When we call the `redo()` method, the Structure is restored to its original state with the value of the key field set to “new value”.

Overall, the History Service is an essential tool for implementing “undo” and “redo” functionality in a UI, allowing users to easily revert or repeat actions as needed.



Structure

The *structure* is the core of the domain model. It represents the deserialized data, which can come in any form thanks to the implementation of the anything pattern.

Type System

The *type system* is another core component of the domain model. It provides the capability to define custom types which is used by extensions to add novel types to the system. Furthermore, the *interpretation service* is responsible for evaluating a structure and determining which types it implements.

Action System

The *action system* is providing the needed infrastructure to define and register actions. Actions are used to define custom procedures which the user can trigger.

Presentation System

The *presentation system* is responsible for everything to do with presentation, including *presenters* and *views*.

Document System

The *document system* is responsible for handling the serialization and deserialization of different *mime types* from various *resource types*.

For a more complete overview of the domain model see the [detailed domain model](#) page.

3.2.3 Extension Development

The development of *extensions* is a vital part of the Black Fennec ecosystem as it allows developers and users to customize and extend the application with their own components and functionality. In this guide, we will cover the basics of extension development for Black Fennec, including how to create and register extensions, how to utilize the Extension API, and how to manage the lifecycle of extensions. By following these guidelines, you will be able to effectively develop and integrate extensions into the Black Fennec application.

In this documentation we will explore the framework for creating, managing, and utilizing extensions in the application. The purpose of the extension system is to allow users to easily extend and customize the Black Fennec application to meet their specific needs and requirements. This can include adding new types, type views, presenters, actions, and other components to the application.

The extension system in Black Fennec consists of several key components, including the *Extension API*, *Extension Service*, *Extension Registry*, and individual *Extensions*. The Extension API is a collection of services that are made available to Extensions and provide access to various features and functionality within the application. The Extension Service is responsible for loading and managing the lifecycle of Extensions, including resolving dependencies and ensuring that Extensions are loaded in the correct order. The Extension Registry is a database of registered Extensions, which is used to store and manage the installed Extensions in the application.

To create an Extension, developers must define their Extension in a Python module and define an `entry_point` in the `setup.py` to be recognized by the Extension Service. Once an Extension is loaded, it can be activated allowing it to access the services and functionality provided by the Extension API.

An example of a minimal Extension is shown in the following text. To read this section it might make sense to open the projects repository which defines a [template for creating extensions](#). This allows you to more efficiently explore the code that is described below.

```
# my_extension.py

from blackfennec.extension_system import Extension
from blackfennec.extension_system import ExtensionApi

class MyExtension(Extension):
    def __init__(self, api: ExtensionApi):
        super().__init__(
            name='My Extension',
            api=api)

def create(api: ExtensionApi) -> MyExtension:
    return MyExtension(api)
```

The Extension class is the base class for all Extensions and provides a number of useful methods and properties that can be used to interact with the application. The `create` function is used to instantiate the Extension and is called by the Extension Service when the Extension is loaded. The `api` parameter is an instance of the Extension API and provides access to the services and functionality provided by the application.

Note: The `create` function must be defined in the module and must return an instance of the Extension class.

In addition to the Python module, an Extension must also define an `entry_point` in the `setup.py` file. This is used by the Extension Service to locate the Extension and load it into the application. The `entry_point` is defined as a `blackfennec.extension` in the `setup.py` file. For example:

```
# setup.py

from setuptools import setup

setup(
    name='my_extension',
    version='0.1.0',
    description='My Extension',
    author='My Name',
    entry_points={
        'blackfennec.extension': [
            'my_extension = my_extension'
        ]
    })
```

Note: The `entry_point` must be defined as a `blackfennec.extension` and must point to the Python module containing the `create` function.

Once the Extension is defined, it can be installed into the application using the `pip` command. For example:

```
$ pip install -e .
```

When the extension is installed in the environment in which Black Fennec is run, it will be loaded by the Extension Service and activated. The Extension Service will then call the `create` function to instantiate the Extension and provide it with an instance of the Extension API. The Extension can then access the services and functionality provided by the Extension API.

Note: The Extension Service will only load Extensions that are installed in the environment in which Black Fennec is run.

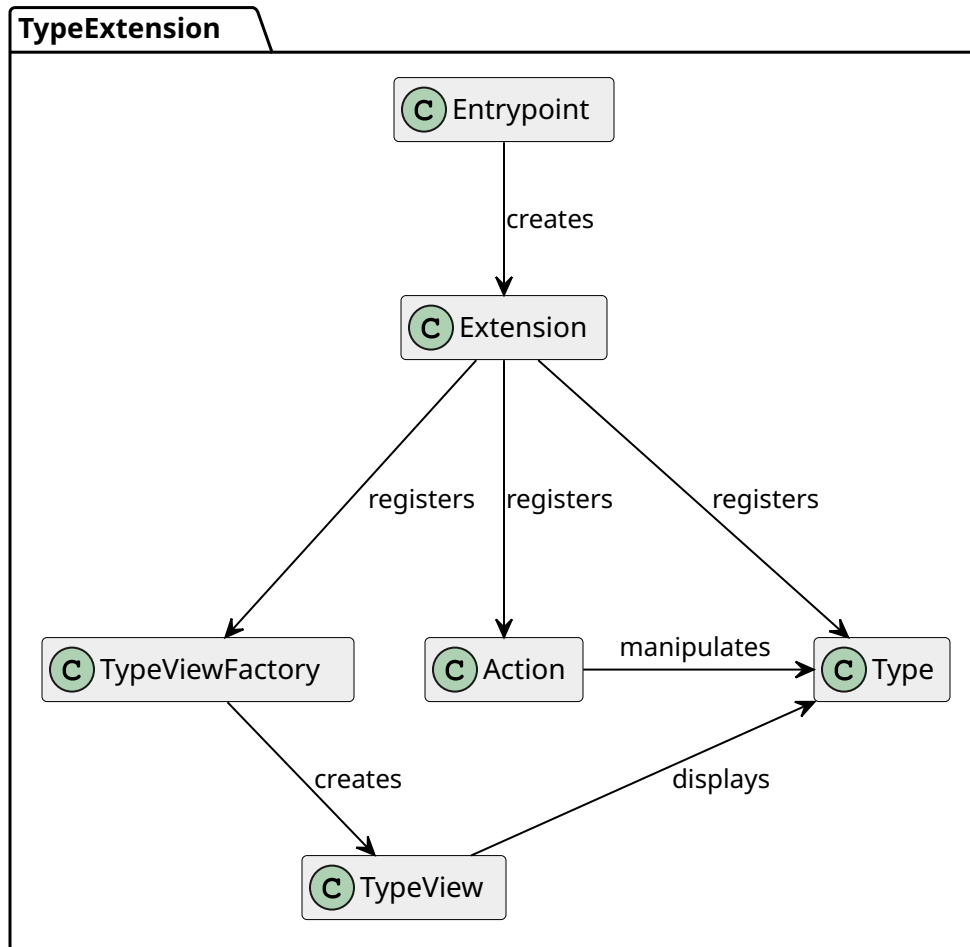
If you run Black Fennec with Flatpak, you must install the Extension in the Flatpak environment.

That's it! You have successfully created and installed an Extension into the Black Fennec application.

Develop a Type Extension

A type extension defines and adds a new *type* `<definition_type>` to the object model. For this new type to be usable it must be accessible to the user. Therefore, it is sensible for the extension to also provide three further components, namely a user interface *Type View* and a factory (*Type View Factory*) that can produce them on demand, and optionally some *Actions*.

Type Extension Overview



Developing a type extension is relatively simple. In this walk-through we develop a *Type*, a custom *Type View* `<type_view>` and an *Action* `<definition_action>` for it.

Note: This walk-through re-implements the `File` type already provided by the `Base` extension. The upstream source code can be found in `extensions/base`.

Defining the Extension Entrypoint

A valid extension is required to provide an entrypoint. Currently this means that the extension must provide a special functions called `create` which takes a single parameter of type *Extension Api*. This function is called when the extension is *loaded*.

For more information on how to create the entrypoint, see the documentation on *Extension Development*.

Creating and Loading a Type Definition

The first file that needs to be created is the Type definition file. This file is a JSON file that contains the definition of the Type and should be named `<type_name>.json`. In the case of a `File` type it could look like this:

```
{
  "super": { "$ref": "bftype://Map" },
  "type": "File",
  "required": [
    "file_path",
    "file_type"
  ],
  "properties": {
    "file_path": {
      "super": null,
      "type": "String"
    },
    "file_type": {
      "super": null,
      "type": "String"
    }
  }
}
```

The Type definition file contains the following information:

- The super class of the Type. This is the Type that our Type extends. In this case the `File` Type extends the *Map Type*. Note, that this is a reference with the `bftype://` protocol. This is a reference to a Type that must already be loaded in the *Type Registry*.
- The name of the Type. This is the name that will be used to reference the Type. In this case the Type is called `File`.
- Required properties. These are the properties that are required to be set on instances of this type. In this case the `file_path` and `file_type` properties are required.
- The properties themselves. These are the properties that are available on the type instance. In this case the `file_path` and `file_type` properties are available.
 - The `file_path` property is a `String` Type.
 - The `file_type` property is a `String` Type.

With this we can extend our Extension to register the `File` Type into the *Type Registry*:

```
from blackfennec.extension_system import Extension
from blackfennec.extension_system import ExtensionApi
```

(continues on next page)

(continued from previous page)

```

class MyExtension(Extension):
    def __init__(self, api: ExtensionApi):
        super().__init__(
            name='My Extension',
            api=api)

        self._file_type = None
        self._action = None

    def register_types(self):
        # currently the type loader does also register the type
        self._file_type = self._api.type_loader.load('file/file.json')

    def deregister_types(self):
        self._api.type_registry.deregister_type(self._file_type)
        self._file_type = None

```

If we didn't want to add any special functionality for our new Type we could stop here.

Creating a Wrapper for the Type

Black Fennec lacks the ability to create a concrete Type instance directly. Instead it is recommended to create a wrapper for the Type that can be used to interact with instances of it. The snippet below is an example of a File wrapper:

```

class File:
    """File type wrapper

    Helper class representing an instance of a 'File'.
    Can be used by other classes as a helper interact with the underlay more easily.
    """

    FILE_PATH_KEY = 'file_path'
    FILE_TYPE_KEY = 'file_type'

    def __init__(self, subject: Map = None):
        """File Constructor

        Args:
            subject (Map): underlying map interpretation to
                which property calls are dispatched
        """
        self._subject: Map = subject or Map()
        if File.FILE_PATH_KEY not in self._subject.value:
            self._subject.add_item(File.FILE_PATH_KEY, String())
        if File.FILE_TYPE_KEY not in self._subject.value:
            self._subject.add_item(File.FILE_TYPE_KEY, String())

    @property
    def subject(self):
        return self._subject

```

(continues on next page)

(continued from previous page)

```

def _get_value(self, key):
    if key not in self.subject.value:
        return None
    return self.subject.value[key].value

def _set_value(self, key, value):
    assert key in self.subject.value
    self.subject.value[key].value = value

@property
def file_path(self) -> str:
    return self._get_value(File.FILE_PATH_KEY)

@file_path.setter
def file_path(self, value: str):
    self._set_value(File.FILE_PATH_KEY, value)

@property
def file_type(self) -> str:
    return self._get_value(File.FILE_TYPE_KEY)

@file_type.setter
def file_type(self, value: str):
    self._set_value(File.FILE_TYPE_KEY, value)

```

Creating the View Model

Next we want to create a view model.

Note: We recommend using MVVM.

```

class FileViewModel:
    """View model for core type File."""

    def __init__(self, interpretation: Interpretation):
        """Create constructor

        Args:
            interpretation (Interpretation): The overarching
            interpretation
        """
        self._interpretation = interpretation
        self._file: File = File(interpretation.structure)

    @property
    def file_path(self):
        """Property for file path"""
        return self._file.file_path

```

(continues on next page)

(continued from previous page)

```

@file_path.setter
def file_path(self, value: str):
    self._file.file_path = value

@property
def file_type(self):
    """Property for file type"""
    return self._file.file_type

@file_type.setter
def file_type(self, value: str):
    self._file.file_type = value

def navigate(self):
    self._interpretation.navigate(self._interpretation.structure)

```

Creating the View

This file depends on how one wants to visualize the Type. Important is that your view is as responsive as possible, as you never know how big a presenter will show your Type View. For an example please see the *FileView* in the Base Extension package located in `extensions/base/base/file`.

Writing a ViewFactory

Creating a view is a non-trivial problem. This is why Black Fennec does not create them itself. Instead you have to register a *ViewFactory* capable of creating a view for your Type.

Luckily creating a view for a File is rather simple. First, we create the view model and after we can construct the appropriate view.

```

class FileViewFactory:
    """Creator of the FileView"""

    def satisfies(self, specification: Specification) -> bool:
        """Test if this view factory can satisfy the specification

        Args:
            specification (Specification): the specification to be satisfied

        Returns:
            bool: True if the specification can be satisfied. Otherwise False.
        """
        return True

    def create(self, interpretation: Interpretation) -> FileView:
        """creates a FileView

        Args:
            interpretation (Interpretation): The overarching
            interpretation.

```

(continues on next page)

(continued from previous page)

```

        specification (Specification): The specification which can fine
            tune the creation function.

    Returns:
        FileView
    """
    view_model = FileViewModel(interpretation)
    if interpretation.specification.is_request_for_preview:
        return FilePreview(view_model)

    return FileView(view_model)

```

Registering the View

The last step is to register the view for the *File Type*. This can be done by adding the following code to the *create_extension* method:

```

class MyExtension(Extension):
    # ...

    def register_view_factories(self):
        self._api.view_registry.register_view_factory(
            self._file_type,
            Specification(),
            FileViewFactory())

    def deregister_view_factories(self):
        self._api.view_registry.deregister_type_view_factory(
            self._file_type,
            Specification())

```

Creating an Action

An *Action* always registers itself to a *Type*. In return we are guaranteed to receive an instance of this type when the action is triggered. We will register our action to the *File type*.

The definition of an action is again relatively simple. We only need to implement a method called *execute* and two properties describing the action.

```

from blackfennec.action_system.action import Action
from blackfennec.action_system.context import Context

class GuessMimeTypeAction(Action):
    def __init__(self,
                 map_type,
                 resource_type_registry: ResourceTypeRegistry):
        super().__init__(map_type)
        self._resource_type_registry: ResourceTypeRegistry = \
            resource_type_registry

```

(continues on next page)

(continued from previous page)

```
def execute(self, context: Context):
    file_type = File(context.structure)

    resource_type_id = ResourceType.try_determine_resource_type(
        file_type.file_path)

    resource_type = self._resource_type_registry.resource_types[
        resource_type_id
    ]

    mime_type = MimeType.try_determine_mime_type(
        file_type.file_path,
        resource_type)
    file_type.file_type = mime_type

@property
def name(self):
    return "guess mime type"

@property
def description(self):
    return """Tries to determine the mime type of a file."""
```

The `execute` method is called when the action is triggered. The `context` parameter contains the instance of the type the action is registered to but we have to “cast” it with our wrapper. In our case this is a `File` instance. We can use this instance to access the file path and to set the mime type.

Finally, we need to register the action to the `File` type. This is done in the `register_actions` method.

```
class MyExtension(Extension):
    # ...

    def register_actions(self):
        self._action = DetectMimeTypeAction(self._file_type)
        self._api.action_registry.register_action(self._action)

    def deregister_actions(self):
        self._api.action_registry.deregister_action(self._action)
        self._action = None
```

Conclusion

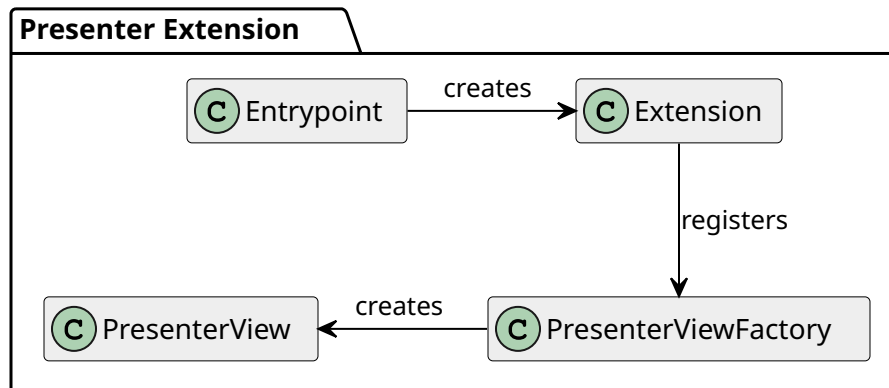
In conclusion, the Black Fennec extension system allows developers to easily add custom functionality to the application through the creation of type extensions. By following the steps outlined in this tutorial, developers can create a new type, create a custom user interface for that type, and add actions to manipulate instances of that type. The extension system provides a clear and intuitive way to extend Black Fennec and tailor it to specific needs and use cases.

Develop a Presenter Extension

Developing a *Presenter extension* certainly is a more advanced task than developing a *Type extensions* as a more detailed understanding of the system is required as for example navigations have to be handled.

As a basis for this walk-through on how to develop a *Presenter extension* we will take the default Presenter of Black Fennec called the column-based-presenter. To read this section it might make sense to open the projects repository located at [here](#). This allows you to individually explore the code lying behind the column-based-presenter while reading through the following explanations.

Presenter Extension Overview



Hint: The structure that is present in this presenter is how Black Fennec recommends an extension should be structured. But as previously mentioned this is up to the developer of the extension.

We will now step by step explain the individual Class required to create a Presenter Extension.

Defining the Extension Entrypoint

A valid extension is required to provide an entrypoint. Currently this means that the extension must provide a special functions called `create` which takes a single parameter of type *Extension Api*. This function is called when the extension is *loaded*.

Based on the entrypoint described as in *Extension Development* we can now implement the `register_presenters` function. This function is called when the extension is loaded and is responsible for registering the presenters that are provided by this extension.

```

1 def register_presenters(self):
2     column_based_presenter = ColumnBasedPresenterViewFactory(
3         self._api.interpretation_service,
  
```

(continues on next page)

(continued from previous page)

```

4         self._api.view_factory,
5     )
6     self._api.presenter_registry.register_presenter(
7         column_based_presenter,
8     )
9     self.presenters.append(column_based_presenter)
10
11 def deregister_presenters(self):
12     for presenter in self.presenters:
13         self._api.presenter_registry.deregister_presenter(type(presenter))

```

As you can see the `extension_api` that is received upon initialization of the extension, can be used to register the presenter and instantiate our presenter factory that we will implement in the next step.

Hint: The *Extension API* is a special object that is provided to the extension upon initialization. It provides access to the services that are provided by the core of Black Fennec.

The *Interpretation Service* is a service that is also provided by the core of Black Fennec and is responsible for interpreting the data that is provided by a document.

Writing a Presenter Factory

The Presenter Factory is a simple factory used to create the presenter view model and inject it into the view. The simple factory is instantiated with all arguments that are later on required by the view model to be able to navigate and interpret information. It is a *must* that the class, in our case the view factory, that is registered in the presenter registry contains a ‘create’ function that receives a instance of the `navigation_service` and a history service.

Hint: The *Navigation Service*. is a service that is provided by the core of Black Fennec and is responsible for navigating between different views.

The *History Service* is a service that is provided by the core of Black Fennec and is responsible for keeping track of the change history of a document.

```

1 class ColumnBasedPresenterViewFactory:
2     """Creator or the ColumnBasedPresenterView"""
3
4     def __init__(self, interpretation_service, view_factory):
5         self._interpretation_service = interpretation_service
6         self._view_factory = view_factory
7
8     def create(self, navigation_service, history) -> ColumnBasedPresenterView:
9         """Create column based presenter view
10
11         Returns:
12             ColumnBasedPresenterView: The column based presenter view.
13             Can be used as presenter in the main UI.
14         """
15         view_model = ColumnBasedPresenterViewModel(
16             self._interpretation_service,

```

(continues on next page)

(continued from previous page)

```
17         navigation_service,  
18         history  
19     )  
20     return ColumnBasedPresenterView(view_model, self._view_factory)
```

As visible in the code example above, the expected return of the create function is the view of our presenter.

Creating a Presenter View

The Presenter View is something, that is in the responsibility of the extension developer. He may inspire himself by looking at the code of the `column-based-presenter - implementation-detail` - which thus not relevant for this walk-through.

Hint: One speciality of black-fennecs implementation of MVVM is the utility class `Observable`, which is a helper class to implement the Observer pattern. With the method call `'bind'`, which is bound via a named-parameter, that is passed a function and will respond to notify events with the event-name that corresponds to the named-parameter, to the view model which inherits of `Observable`.

The Presenter View Model

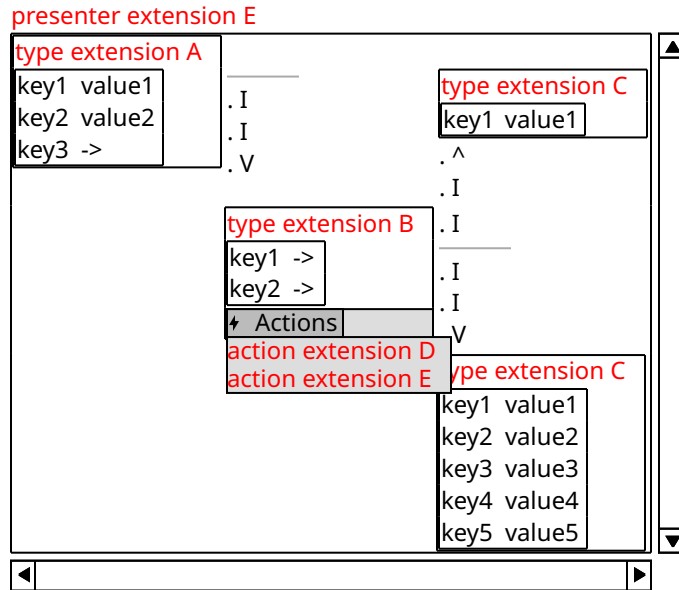
The presenter that is currently active in Black Fennec gets notified by a Black Fennec component via the `'show'` function. This function gets passed which interpretation has triggered the show event, and which part of the *structure* should now be displayed. This structure can be interpreted with the *interpretation_service* in order for types beyond the `core_types` to be shown.

Hint: It is the responsibility of the presenter of setting the navigation service on the interpretation he created. Otherwise, navigational requests that happen in the interpretation would not reach the presenter.

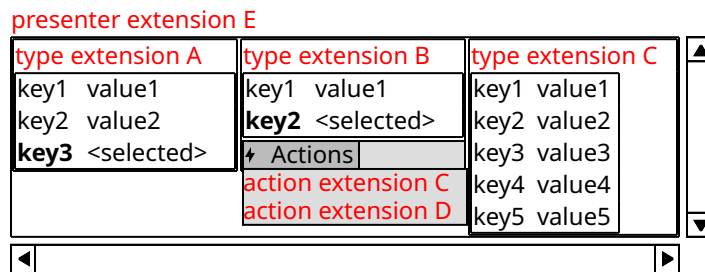
Presenter Extension

The presenter extension (a.k.a Structure Presenter Extension) is responsible for displaying and positioning all Structure Views as described in *Develop a Type Extension* as well as making *Actions* <definition_action> available to the user. Presenters have few restrictions and will be given a rectangular area for rendering.

Wireframe of a Graph Based Presenter



Wireframe of a Column Based Presenter



Displaying Structure Views

Structure Views are placed and positioned by the presenter. They ought to expect a rectangular region in which they are allowed to render information. The presenter has full control over size and position of these regions and is free to arrange and decorate them. It is not the responsibility of the presenter to ensure the usability of the Structure View. However, we might define a soft limit to the minimum width or height that a Structure View ought to get.

Disclaimer

At this point we do NOT plan on implementing more than one presenter. This means that there exists no selection possibility for the user. The presenter that is used is the one that is registered in the *presenter registry* first.

3.2.4 Testing

To ensure the stability, usability, and functional correctness of the system, we deploy numerous testing strategies including usability, system, integration, and unit tests. The unit and integration test are written in Python and executed with the pytest framework during the CI/CD pipeline whilst the usability tests are conducted manually on every major or minor release. The usability tests are conducted after every major release.

The pages below document the results of the different manually conducted tests.

System Tests

Release v0.12.0

Nr.	Functional Area	Test Name	Non-Functional Requirements	Test Steps	Expected Results	Actual Results	Test passed / failed	Tester, Time, Date
1	Main UI	Run Black Fennec	<ul style="list-style-type: none"> 2.2.1. Correctness 2.2.3.1.1. Fast Starter 2.2.5.2.1. Just Like an Apple 	<ol style="list-style-type: none"> 1. Click 'run Black Fennec' 	<ol style="list-style-type: none"> 1. Splash screen appears within 500ms 2. Splash screen disappears 3. Main ui appears 	<ol style="list-style-type: none"> 1. Splash screen not visible 2. Splash screen not visible 3. Main ui appears 	passed	S. Kindhauser, 12:23, 2022-12-23
2	Main UI	close/quit Black Fennec	<ul style="list-style-type: none"> 2.2.1. Correctness 2.2.5.2.1. Just Like an Apple 	<ol style="list-style-type: none"> 1. Click cross button 	<ol style="list-style-type: none"> 1. Black Fennec closes itself, all related windows disappear 	<ol style="list-style-type: none"> 1. Black Fennec closes itself, all related windows disappear 	passed	S. Kindhauser, 12:25, 2022-12-23
3	Main UI	Open new Project	<ul style="list-style-type: none"> 2.2.1. Correctness 2.2.3.1.2. Loading Projects 	<ol style="list-style-type: none"> 1. Click menu in the toolbar 2. Click 'Open Directory' 	<ol style="list-style-type: none"> 1. dropdown menu appears 2. file dialogue window pops up 3. file 	<ol style="list-style-type: none"> 1. dropdown menu appears 2. file dialogue window pops up 3. file 	passed	S. Kindhauser, 12:27, 2022-12-23
60			<ul style="list-style-type: none"> 2.2.4. Work in Extor 	<ol style="list-style-type: none"> 2.1. Browse 	<ol style="list-style-type: none"> 3. file 	<ol style="list-style-type: none"> 3. file 	Chapter 3. Project	

Release v0.11.0

Nr.	Functional Area	Test Name	Non-Functional Requirements	Test Steps	Expected Results	Actual Results	Test passed / failed	Tester, Time, Date
1	Main UI	Run Black Fennec	<ul style="list-style-type: none"> 2.2.1.2. Correctness 2.2.3.1.1. Fast Starter 2.2.5.2.1. Just Like an Apple 	<ol style="list-style-type: none"> 1. Click 'run Black Fennec' 	<ol style="list-style-type: none"> 1. Splash screen appears within 500ms 2. Splash screen disappears 3. Main ui appears 	<ol style="list-style-type: none"> 1. Splash screen not visible 2. Splash screen not visible 3. Main ui appears 	passed	S. Kindhauser, 15:23, 2022-12-14
2	Main UI	close/quit Black Fennec	<ul style="list-style-type: none"> 2.2.1.2. Correctness 2.2.5.2.1. Just Like an Apple 	<ol style="list-style-type: none"> 1. Click cross button 	<ol style="list-style-type: none"> 1. Black Fennec closes itself, all related windows disappear 	<ol style="list-style-type: none"> 1. Black Fennec closes itself, all related windows disappear 	passed	S. Kindhauser, 15:25, 2022-12-14
3	Main UI	Open new Project	<ul style="list-style-type: none"> 2.2.1.2. Correctness 2.2.3.1.2. Loading Projects 	<ol style="list-style-type: none"> 1. Click menu in the toolbar 2. Click 'Open Directory' 	<ol style="list-style-type: none"> 1. dropdown menu appears 2. file dialogue window pops up 3. file 	<ol style="list-style-type: none"> 1. dropdown menu appears 2. file dialogue window pops up 3. file 	passed	S. Kindhauser, 15:27, 2022-12-14
62			<ul style="list-style-type: none"> 2.2.4.2.1. Work in Extor 	<ol style="list-style-type: none"> 1. Browse 	<ol style="list-style-type: none"> 3. file 	<ol style="list-style-type: none"> 3. file 	Chapter 3. Project	

Release v0.10.0

Nr.	Functional Area	Test Name	Non-Functional Requirements	Test Steps	Expected Results	Actual Results	Test passed / failed	Tester, Time, Date
1	Main UI	Run Black Fennec	<ul style="list-style-type: none"> 2.2.1. Correctness 2.2.3.1.1. Fast Starter 2.2.5.2.1. Just Like an Apple 	<ol style="list-style-type: none"> 1. Click 'run Black Fennec' 	<ol style="list-style-type: none"> 1. Splash screen appears within 500ms 2. Splash screen disappears 3. Main ui appears 	<ol style="list-style-type: none"> 1. Splash screen not visible 2. Splash screen not visible 3. Main ui appears 	passed	C. Martens, 17:23, 2022-11-22
2	Main UI	close/quit Black Fennec	<ul style="list-style-type: none"> 2.2.1. Correctness 2.2.5.2.1. Just Like an Apple 	<ol style="list-style-type: none"> 1. Click cross button 	<ol style="list-style-type: none"> 1. Black Fennec closes itself, all related windows disappear 	<ol style="list-style-type: none"> 1. Black Fennec closes itself, all related windows disappear 	passed	C. Martens, 17:25, 2022-11-22
3	Main UI	Open new Project	<ul style="list-style-type: none"> 2.2.1. Correctness 2.2.3.1.2. Loading Projects 	<ol style="list-style-type: none"> 1. Click menu in the toolbar 2. Click 'Open Directory' 	<ol style="list-style-type: none"> 1. dropdown menu appears 2. file dialogue window pops up 3. file 	<ol style="list-style-type: none"> 1. dropdown menu appears 2. file dialogue window pops up 3. file 	passed	C. Martens, 17:27, 2022-11-22
64			<ul style="list-style-type: none"> 2.2.4. Work in Extor 	<ol style="list-style-type: none"> 2.1. Browse 	<ol style="list-style-type: none"> 3. file 	<ol style="list-style-type: none"> 3. file 	Chapter 3. Project	

Release v0.9

Nr.	Functional Area	Test Name	Non-Functional Requirements	Test Steps	Expected Results	Actual Results	Test passed / failed	Tester, Time, Date
1	Main UI	Run Black Fennec	<ul style="list-style-type: none"> 2.2.1. Correctness 2.2.3.1.1. Fast Starter 2.2.5.2.1. Just Like an Apple 	<ol style="list-style-type: none"> 1. Click 'run Black Fennec' 	<ol style="list-style-type: none"> 1. Splash screen appears within 500ms 2. Splash screen disappears 3. Main ui appears 	<ol style="list-style-type: none"> 1. Splash screen not visible 2. Splash screen not visible 3. Main ui appears 	passed	S. Kindhauser, 11:23, 2022-11-04
2	Main UI	close/quit Black Fennec	<ul style="list-style-type: none"> 2.2.1. Correctness 2.2.5.2.1. Just Like an Apple 	<ol style="list-style-type: none"> 1. Click cross button 	<ol style="list-style-type: none"> 1. Black Fennec closes itself, all related windows disappear 	<ol style="list-style-type: none"> 1. Black Fennec closes itself, all related windows disappear 	passed	S. Kindhauser, 11:25, 2022-11-04
3	Main UI	Open new Project	<ul style="list-style-type: none"> 2.2.1. Correctness 2.2.3.1.2. Loading Projects 	<ol style="list-style-type: none"> 1. Click menu in the toolbar 2. Click 'Open Directory' 	<ol style="list-style-type: none"> 1. dropdown menu appears 2. file dialogue window pops up 3. file 	<ol style="list-style-type: none"> 1. dropdown menu appears 2. file dialogue window pops up 3. file 	passed	S. Kindhauser, 11:27, 2022-11-04
66			<ul style="list-style-type: none"> 2.2.4. Work in Extor 	<ol style="list-style-type: none"> 2.1. 3. Brows 	<ol style="list-style-type: none"> 3. file 	<ol style="list-style-type: none"> 3. file 	Chapter 3. Project	

Release v0.8

Nr.	Functional Area	Test Name	Non-Functional Requirements	Test Steps	Expected Results	Actual Results	Test passed / failed	Tester, Time, Date
1	Main UI	Run Black Fennec	<ul style="list-style-type: none"> 2.2.1.2. Correctness 2.2.3.1.1. Fast Starter 2.2.5.2.1. Just Like an Apple 	<ol style="list-style-type: none"> 1. Click 'run Black Fennec' 	<ol style="list-style-type: none"> 1. Splash screen appears within 500ms 2. Splash screen disappears 3. Main ui appears 	<ol style="list-style-type: none"> 1. Splash screen appears within 500ms 2. Splash screen disappears 3. Main ui appears 	passed	S. Kindhauser, 11:24, 2022-10-11
2	Main UI	close/quit Black Fennec	<ul style="list-style-type: none"> 2.2.1.2. Correctness 2.2.5.2.1. Just Like an Apple 	<ol style="list-style-type: none"> 1. Click cross button 	<ol style="list-style-type: none"> 1. Black Fennec closes itself, all related windows disappear 	<ol style="list-style-type: none"> 1. Black Fennec closes itself, all related windows disappear 	passed	S. Kindhauser, 11:26, 2022-10-11
3	Main UI	Open new Project	<ul style="list-style-type: none"> 2.2.1.2. Correctness 2.2.3.1.2. Loading Projects 2.2.4.2.1. Work in External 	<ol style="list-style-type: none"> 1. Click 'Project' in the toolbar 2. Click 'Open' in drop-down menu 3. Brows 	<ol style="list-style-type: none"> 1. drop-down menu appears 2. file dialogue window pops up 3. file di- 	<ol style="list-style-type: none"> 1. drop-down menu appears 2. file dialogue window 3. file di- 	passed	S. Kindhauser, 11:29, 2022-10-11
68							Chapter 3. Project	

Release v0.6

Nr.	Functional Area	Test Name	Non-Functional Requirements	Test Steps	Expected Results	Actual Results	Test passed / failed	Tester, Time, Date
1	Main UI	Run Black Fennec	<ul style="list-style-type: none"> 2.2.1.2. Correctness 2.2.3.1.1. Fast Starter 2.2.5.2.1. Just Like an Apple 	<ol style="list-style-type: none"> 1. Click 'run Black Fennec' 	<ol style="list-style-type: none"> 1. Splash screen appears within 500ms 2. Splash screen disappears 3. Main ui appears 	<ol style="list-style-type: none"> 1. Splash screen appears within 500ms 2. Splash screen disappears 3. Main ui appears 	passed	Lara Gubler, 22:24, 25.05.2021
2	Main UI	close/quit Black Fennec	<ul style="list-style-type: none"> 2.2.1.2. Correctness 2.2.5.2.1. Just Like an Apple 	<ol style="list-style-type: none"> 1. Click cross button 	<ol style="list-style-type: none"> 1. Black Fennec closes itself, all related windows disappear 	<ol style="list-style-type: none"> 1. Black Fennec closes itself, all related windows disappear 	passed	Lara Gubler, 22:26, 25.05.2021
3	Main UI	Open new Project	<ul style="list-style-type: none"> 2.2.1.2. Correctness 2.2.3.1.2. Loading Projects 2.2.4.2.1. Work in External 	<ol style="list-style-type: none"> 1. Click 'Project' in the toolbar 2. Click 'Open' in drop-down menu 3. Brows 	<ol style="list-style-type: none"> 1. drop-down menu appears 2. file dialogue window pops up 3. file di- 	<ol style="list-style-type: none"> 1. drop-down menu appears 2. file dialogue window 3. file di- 	passed	Lara Gubler, 22:29, 25.05.2021
70							Chapter 3. Project	

Release v0.5

Nr.	Functional Area	Test Name	Test Steps	Expected Results	Actual Results	Test passed / failed	Non-Functional Requirements	Tester, Time, Date
1	Main UI	Run Black Fennec	1. Click 'run Black Fennec'	1. Splash screen appears within 500ms 2. Splash screen disappears 3. Main ui appears	1. Splash screen appears within 500ms 2. Splash screen disappears 3. Main ui appears	passed	<ul style="list-style-type: none"> 2.2.1.2 Correctness 2.2.3.1.1 Fast Starter 2.2.5.2.1 Just Like an Apple 	Lara Gubler, 22:00, 2021.05.18
2	Main UI	close/quit Black Fennec	1. Click cross button	1. Black Fennec closes itself, all related windows disappear	1. Black Fennec closes itself, all related windows disappear	passed	<ul style="list-style-type: none"> 2.2.1.2 Correctness 2.2.5.2.1 Just Like an Apple 	Lara Gubler, 22:04, 2021.5.18
3	Main UI	Open new Project	1. Click 'Project' in the toolbar 2. Click 'Open' in drop-down menu	1. drop-down menu appears 2. file dialogue window pops up 3. file dialogue window appears	1. drop-down menu appears 2. file dialogue window pops up 3. file dialogue window appears	passed	<ul style="list-style-type: none"> 2.2.1.2 Correctness 2.2.3.1.2 Loading Projects 2.2.4.2.1 Work in External 	Lara Gubler, 10:14, 2021.05.18
72							Chapter 3. Project	

Release v0.3

Nr.	Functional Area	Test Name	Test Steps	Expected Results	Actual Results	Test passed / failed	Non-Functional Requirements	Tester, Time, Date
1	Main UI	Run Black Fennec	1. Click 'run Black_Fennec'	1. Splash screen appears within 500ms 2. Splash screen disappears 3. Main ui appears	1. Splash screen appears within 500ms 2. Splash screen disappears 3. Main ui appears	passed	<ul style="list-style-type: none"> 2.2.1.2. Correctness 2.2.3.1.1. Fast Starter 	Leonie Däullary, 10:30, 2021.04.09
2	Main UI	close/quit Black Fennec	1. Click cross button	1. Black Fennec closes itself, all related windows disappear	1. Black Fennec closes itself, all related windows disappear	passed	<ul style="list-style-type: none"> 2.2.1.2. Correctness 	Leonie Däullary, 10:30, 2021.04.09
3	Main UI	Open Json file	1. Click 'Open' in toolbar 2. Click 'Open' in drop-down menu	1. drop-down menu appears 2. file dialogue window pops up 3. file dialogue window	1. drop-down menu appears 2. file dialogue window pops up 3. file dialogue window	passed	<ul style="list-style-type: none"> 2.2.1.2. Correctness 2.2.3.1.2. Loading Projects 2.2.4.2.1. Work in External 	Leonie Däullary, 10:35, 2021.04.09
74							Chapter 3. Project	

Usability Study Release v0.6

This usability test was conducted for the release of version 0.6.0. It contains multiple tasks that can be used to the systems usability. These tasks are dedicated to personas that have little to none experience with the application, in order to be representative of users that use our application for the first time.

Personas

In this study four persons participated. We mapped three of the testers to a *persona*, which fits their background and potential usage best. The last tester was not in the target group and had a non technical background.

Tester	Persona
CC	<i>Matthew G. Gritton</i>
YH	<i>Sun Chuang</i>
MB	<i>Alan Nordin</i>
CH	Not in target group

Tasks

The tasks are to be read to the user that is testing the application. The tester is encouraged to think aloud by the test supervisor, by asking them to whenever necessary.

Open a Project

Open the project *examples* which is located under *\$BLACK_FENNEC_HOME/examples*.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.

Goals

The project is loaded and the associated files are accessible through the *Project File Tree*.

Observation

Tester	Observation
CC	Intuitively Correct
YH	Intuitively Correct
MB	Intuitively Correct
CH	Double clicked folders, and was unsure how to open project, because in folder with only files nothing could be selected.

Open a File

Open the file named *black_fennec.json*.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.

Goals

The presenter displays the file.

Observation

Tester	Observation
CC	Intuitively Correct
YH	Intuitively Correct
MB	Intuitively Correct
CH	Single click tried first, then double click

Navigate Structure

In the file *black_fennec.json*, find and navigate to the *Person* called *Lara*.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* is open.

Goals

The user recognises the correct location.

Observation

Tester	Observation
CC	N/A
YH	N/A
MB	Intuitively Correct; List is not immediately obvious.
CH	Intuitively Correct; Thinks whole list item should be clickable

Open two Files

Open the files named *black_fennec.json* and *user_db.json*.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.

Goals

The main ui opened two tabs, each with a presenter displaying one file.

Observation

Tester	Observation
CC	Intuitively Correct; Did not expect file to open in background.
YH	Intuitively Correct; Did not expect file to open in background.
MB	Intuitively almost Correct; Did not immediately see newly opened file and reopened file; Did not expect file to open in background.
CH	Intuitively Correct; Did not expect file to open in background.

Switch Tabs

Switch between to opened files.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- Two files have been opened.

Goals

The content of the two files can be displayed at will.

Observation

Tester	Observation
CC	Intuitively Correct
YH	Intuitively Correct
MB	Intuitively Correct
CH	Intuitively Correct

Edit Text

Edit the text in a text field.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* has been opened.

Goals

The text in any text field has changed.

Observation

Tester	Observation
CC	Intuitively Correct
YH	Intuitively Correct
MB	Intuitively Correct
CH	Intuitively Correct; Noticed that language differed when right clicking text field

Edit Truth Value

Edit the value of a true/false question.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* has been opened.

Goals

The truth value in any switch has changed.

Observation

Tester	Observation
CC	Intuitively Correct
YH	Intuitively Correct
MB	Intuitively Correct
CH	Intuitively Correct; Not immediately clear what was meant with truth value

Edit Field Name

Edit the name of a filed/row.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* has been opened.
- A map is presented to the user.

Goals

The key in a map item has changed.

Observation

Tester	Observation
CC	Intuitively almost Correct; first tried Double Click, then Right Click; Expected current key value in the text field
YH	Intuitively almost Correct; first tried Singe Click, then Double Click then Right Click; Did not know which row he is editing; Expected row to be highlighted
MB	Intuitively almost Correct; Did not intuitively expect key to be editable; First tried Double Click, then Right Click; No further comments
CH	Intuitively Correct

Remove Field

Remove a filed/row.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* has been opened.
- A map or list is presented to the user.

Goals

The row in a map/list item is removed.

Observation

Tester	Observation
CC	Intuitively Correct
YH	Intuitively Correct
MB	Intuitively Correct
CH	Intuitively Correct

Add Field to List

Add a item of type *String* to a list.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* has been opened.
- A list is presented to the user.

Goals

A row in a list item of type *String* was added.

Observation

Tester	Observation
CC	Intuitively almost Correct; Expected Right Click in empty Space; Expected Add Button;
YH	Intuitively almost Correct; Expected Right Click in empty Space; Expected Add Button; First entered random text, then inspects dropdown menu; Expected behaviour not clearly communicated.
MB	Intuitively almost Correct; Purpose of Template Text Filed not intuitively clear
CH	Intuitively almost Correct; Expected Right Click in empty Space; Expected Add Button;

Add Field to Map

Add a item of type *String* to a map.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* has been opened.
- A map is presented to the user.

Goals

A row in a map item of type *String* was added.

Observation

Tester	Observation
CC	Intuitively Correct; Same issues as with <i>Add Field to List</i>
YH	Intuitively Correct; Same issues as with <i>Add Field to List</i>
MB	Intuitively Correct; Same issues as with <i>Add Field to List</i>
CH	Intuitively Correct; Same issues as with <i>Add Field to List</i>

Save File

Save changes made to a file.

Preconditions

Black Fennec is started. - The main UI is loaded. - Presenter is configured. - The project *examples* is open. - A file is open. - The file was edited.

Goals

Changes made to file are persisted.

Observation

Tester	Observation
CC	Intuitively Correct; Expected Auto Save; No Feedback on Action; Unexpected behaviour: saves all files => Rename button to “save all”?
YH	Intuitively Correct
MB	Intuitively Correct; Expected entire project to be saved (which is what happens); Expected Feedback on Action
CH	Intuitively Correct

Usability Study Release v0.11

This usability test was conducted for version 0.11.0. This is the release preceding version 1.0 which optimally can address some of the recognized shortcomings.

It contains multiple tasks that can be used to the the systems usability. These tasks are dedicated to personas that have little to none experience with the application, in order to be representative of users that use our application for the first time.

Personas

2 personas have participated in this study. We mapped each of them to a *persona* which most closely fits their background and potential usage.

Tester	Persona
TW	<i>Alan Nordin</i>
MB	<i>Sun Chuang</i>

Tasks

The tasks are to be read to the user that is testing the application. The tester is encouraged to think aloud by the test supervisor, by asking them to whenever necessary.

Open a Project

Open the folder *examples* which is located under *\$BLACK_FENNEC_HOME/examples*.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.

Goals

The project is loaded and the associated files are accessible through the *Project File Tree*.

Observation

Tester	Observation
TW	Intuitively correct
MB	Intuitively correct; suggested to remove empty list pattern

Open a File

Open the file named *black_fennec.json*.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.

Goals

The presenter displays the file.

Observation

Tester	Observation
TW	Intuitively correct
MB	Intuitively correct; expected context menu in file tree (e.g. rename file)

Navigate Structure

In the file *black_fennec.json*, find and navigate to the *Person* called *Lara*.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* is open.

Goals

The user recognises the correct location.

Observation

Tester	Observation
TW	“empty” list is not intuitive; suggests showing a preview. Is concerned that new columns might be missed by user; could be solved with auto-scrolling.
MB	“empty” list is not intuitive; suggests showing a preview.

Open two Files

Open the files named *black_fennec.json* and *user_db.json*.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.

Goals

The main ui opened two tabs, each with a presenter displaying one file.

Observation

Tester	Observation
TW	Intuitively correct
MB	Intuitively correct; expected tab switch when “opening” same file multiple times

Switch Tabs

Switch between to opened files.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- Two files have been opened.

Goals

The content of the two files can be displayed at will.

Observation

Tester	Observation
TW	Intuitively correct
MB	Intuitively correct

Edit Text

Edit the text in a text field.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* has been opened.

Goals

The text in any text field has changed.

Observation

Tester	Observation
TW	Intuitively correct. Not sure if the change is saved automatically; suggests change indicator (e.g. asterisk) and a warning on close.
MB	Intuitively correct; expects save button or change indicator

Edit Truth Value

Edit the value of a true/false question.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* has been opened.

Goals

The truth value in any switch has changed.

Observation

Tester	Observation
TW	Intuitively correct
MB	Intuitively correct

Edit Field Name

Edit the name of a field/row.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* has been opened.
- A map is presented to the user.

Goals

The key in a map item has changed.

Observation

Tester	Observation
TW	double click, right click, edit button
MB	right click, edit button; expected check mark to save entire file

Remove Field

Remove a field/row.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* has been opened.
- A map or list is presented to the user.

Goals

The row in a map/list item is removed.

Observation

Tester	Observation
TW	first tried using the delete key, then edit button; noted it to be intuitive but would like to see more short cuts like this.
MB	Intuitively correct

Add Field to List

Add a item of type *String* to a list.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* has been opened.
- A list is presented to the user.

Goals

A row in a list item of type *String* was added.

Observation

Tester	Observation
TW	Intuitively correct
MB	expected enter to add item. entered unknown type in type field; was not stopped from doing so.

Add Field to Map

Add a item of type *String* to a map.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* has been opened.
- A map is presented to the user.

Goals

A row in a map item of type *String* was added.

Observation

Tester	Observation
TW	Intuitively correct
MB	Intuitively correct

Save File

Save changes made to a file.

Preconditions

Black Fennec is started. - The main UI is loaded. - Presenter is configured. - The project *examples* is open. - A file is open. - The file was edited.

Goals

Changes made to file are persisted.

Observation

Tester	Observation
TW	Intuitively correct; right click on tab
MB	Intuitively correct; expected 'save changes' warning on close

Run an Action

Run an action.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- A file is open.

Goals

The user finds the action and can run it.

Observation

Tester	Observation
TW	Intuitively correct
MB	Expected to run action from field of string

Open File from File Manager

From the file manager, open a file in Black Fennec.

Preconditions

- Black Fennec is NOT started.
- The file manager displays a JSON file.

Goals

The file is opened in Black Fennec when double clicked.

Observation

Tester	Observation
TW	Intuitively correct
MB	Intuitively correct

Missing Extension Dialog

Install the missing recommended extensions.

Preconditions

- Black Fennec is installed correctly.
- A recommended extension is missing.
- Black Fennec is started.
- The missing extensions dialog is displayed.

Goals

The recommended extensions are installed.

Observation

Tester	Observation
TW	Intuitively correct and restarted Black Fennec.
MB	Did not read error fully, expected 'install' to just install the missing extensions, did not immediately find extensions in software center. Suggested better error message and to rename the 'install' button to 'open software center'

The missing recommended extensions dialog does not inform the user that Black Fennec must be restarted to use newly installed extensions.

Notes

Some of the subjects wished for a more keyboard friendly UX. The argument is, that users are more productive with keyboard shortcuts. Some of the suggested shortcuts are:

- Delete: Delete a row in a list or map
- Keyboard navigation through the structure

It was also noted, that changing the key of a map item is somewhat cumbersome as the user has to click on the edit button located at the very top of the column. Suggested alternatives include:

- Double click on the key
- Right click on the row

It was also noted that users tend to ignore error messages and just click the most suggestive button. We should consider to change the error message to something more informative and to rename the button to something more descriptive of what actually happens.

Conclusion

Although the test subjects were not familiar with Black Fennec, they were able to use it mostly intuitively. However, the study has also shown that there is room for improvement.

Test Templates

This section contains templates for conducting system and UX tests. These templates are intended to be used as a template for testing and should be extended when new system or UX features are implemented.

System Test Templates

Nr.	Functional Area	Test Name	Non-Functional Requirements	Test Steps	Expected Results	Actual Results	Test failed /	Tester, Time, Date
1	Main UI	Run Black Fennec	<ul style="list-style-type: none"> 2.2.1.2. Correctness 2.2.3.1.1. Fast Starter 2.2.5.2.1. Just Like an Apple 	<ol style="list-style-type: none"> 1. Click 'run Black Fennec' 	<ol style="list-style-type: none"> 1. Splash screen appears within 500ms 2. Splash screen disappears 3. Main ui appears 	<ol style="list-style-type: none"> 1. Splash screen not visible 2. Splash screen not visible 3. Main ui appears 		
2	Main UI	close/quit Black Fennec	<ul style="list-style-type: none"> 2.2.1.2. Correctness 2.2.5.2.1. Just Like an Apple 	<ol style="list-style-type: none"> 1. Click cross button 	<ol style="list-style-type: none"> 1. Black Fennec closes itself, all related windows disappear 	<ol style="list-style-type: none"> 1. Black Fennec closes itself, all related windows disappear 		
3	Main UI	Open new Project	<ul style="list-style-type: none"> 2.2.1.2. Correctness 2.2.3.1.2. Loading Projects 	<ol style="list-style-type: none"> 1. Click menu in the toolbar 2. Click 'Open Directory' 	<ol style="list-style-type: none"> 1. dropdown menu appears 2. file dialogue window pops up 3. file 	<ol style="list-style-type: none"> 1. dropdown menu appears 2. file dialogue window pops up 3. file 		
94			<ul style="list-style-type: none"> 2.2.4.2.1. Work in Explorer 	<ol style="list-style-type: none"> 1. Click menu in the toolbar 2. Click 'Open Directory' 3. Browse 	<ol style="list-style-type: none"> 1. dropdown menu appears 2. file dialogue window pops up 3. file 	<ol style="list-style-type: none"> 1. dropdown menu appears 2. file dialogue window pops up 3. file 	Chapter 3. Project	

Usability Test Templates

This template can be used to conduct usability tests of the Black Fennec application. It contains multiple tasks that can be used to test the system's usability. These tasks are dedicated to personas that have little to no experience with the application, in order to be representative of users that use our application for the first time.

Tasks

The tasks are to be read to the user that is testing the application. The tester is encouraged to think aloud by the test supervisor, by asking them to whenever necessary.

Open a Project

Open the project *examples* which is located under *\$BLACK_FENNEC_HOME/examples*.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.

Goals

The project is loaded and the associated files are accessible through the *Project File Tree*.

Observation

Tester	Observation
<Name>	<Observation>

Open a File

Open the file named *black_fennec.json*.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.

Goals

The presenter displays the file.

Observation

Tester	Observation
<Name>	<Observation>

Navigate Structure

In the file *black_fennec.json*, find and navigate to the *Person* called *Lara*.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* is open.

Goals

The user recognises the correct location.

Observation

Tester	Observation
<Name>	<Observation>

Open two Files

Open the files named *black_fennec.json* and *user_db.json*.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.

Goals

The main ui opened two tabs, each with a presenter displaying one file.

Observation

Tester	Observation
<Name>	<Observation>

Switch Tabs

Switch between to opened files.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- Two files have been opened.

Goals

The content of the two files can be displayed at will.

Observation

Tester	Observation
<Name>	<Observation>

Edit Text

Edit the text in a text field.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* has been opened.

Goals

The text in any text field has changed.

Observation

Tester	Observation
<Name>	<Observation>

Edit Truth Value

Edit the value of a true/false question.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* has been opened.

Goals

The truth value in any switch has changed.

Observation

Tester	Observation
<Name>	<Observation>

Edit Field Name

Edit the name of a filed/row.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* has been opened.
- A map is presented to the user.

Goals

The key in a map item has changed.

Observation

Tester	Observation
<Name>	<Observation>

Remove Field

Remove a filed/row.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* has been opened.
- A map or list is presented to the user.

Goals

The row in a map/list item is removed.

Observation

Tester	Observation
<Name>	<Observation>

Add Field to List

Add a item of type *String* to a list.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* has been opened.
- A list is presented to the user.

Goals

A row in a list item of type *String* was added.

Observation

Tester	Observation
<Name>	<Observation>

Add Field to Map

Add a item of type *String* to a map.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- The project *examples* is open.
- The file *black_fennec.json* has been opened.
- A map is presented to the user.

Goals

A row in a map item of type *String* was added.

Observation

Tester	Observation
<Name>	<Observation>

Save File

Save changes made to a file.

Preconditions

Black Fennec is started. - The main UI is loaded. - Presenter is configured. - The project *examples* is open. - A file is open. - The file was edited.

Goals

Changes made to file are persisted.

Observation

Tester	Observation
<Name>	<Observation>

Run an Action

Run an action.

Preconditions

- Black Fennec is started.
- The main UI is loaded.
- Presenter is configured.
- A file is open.

Goals

The user finds the action and can run it.

Observation

Tester	Observation
<Name>	<Observation>

Open File from File Manager

From the file manager, open a file in Black Fennec.

Preconditions

- Black Fennec is NOT started.
- The file manager displays a JSON file.

Goals

The file is opened in Black Fennec when double clicked.

Observation

Tester	Observation
<Name>	<Observation>

Missing Extension Dialog

Install the missing recommended extensions.

Preconditions

- Black Fennec is installed correctly.
- A recommended extension is missing.
- Black Fennec is started.
- The missing extensions dialog is displayed.

Goals

The recommended extensions are installed.

Observation

Tester	Observation
<Name>	<Observation>

The missing recommended extensions dialog does not inform the user that Black Fennec must be restarted to use newly installed extensions.

Notes

Add notes here.

Conclusion

Add conclusion here.

3.2.5 Code Documentation

The code documentation is generated from the docstrings in the code.

blackfennec

blackfennec package

Subpackages

blackfennec.action_system package

Submodules

blackfennec.action_system.action module

class blackfennec.action_system.action.**Action**(*type: Type*)

Bases: object

Action Interface

This is the interface for all actions.

type

type which is used to identify the action

abstract property description: str

An informative description of the action.

abstract execute(*context: Context*) → None

Function to execute the action

Parameters

context (*Context*) – context of the action

abstract property name: str

A short, identifying name for the action.

blackfennec.action_system.action_registry module

class blackfennec.action_system.action_registry.**ActionRegistry**

Bases: object

property actions: dict[*blackfennec.type_system.type.Type*,
list[*blackfennec.action_system.action.Action*]]

deregister_action(*action: Action*) → None

Function to deregister an action

get_actions(*type: Type*) → list[*blackfennec.action_system.action.Action*]

Function to get all actions for a specific type

register_action(*action: Action*) → None

Function to register a new action

blackfennec.action_system.context module

class blackfennec.action_system.context.**Context**(*ui_context, structure: Structure = None*)

Bases: object

Context is the context in which an action is called.

structure

the structure on which the action is called

Type

Structure

Module contents

`blackfennec.document_system` package

Subpackages

`blackfennec.document_system.mime_type` package

Subpackages

`blackfennec.document_system.mime_type.in_memory` package

Submodules

`blackfennec.document_system.mime_type.in_memory.in_memory_mime_type` module

class

`blackfennec.document_system.mime_type.in_memory.in_memory_mime_type.InMemoryMimeType`

Bases: *MimeType*

export_structure(*output: IO, structure: Structure*)

Exports a structure to a IO Stream

Parameters

- **output** (*IO*) – The IO Stream to export to
- **structure** (*Structure*) – The structure to export

import_structure(*to_import*) → *Structure*

Imports a structure from a IO Stream

Parameters

to_import (*IO*) – The IO Stream to import from

Returns

The imported structure

Return type

Structure

property mime_type_id: `str`

Returns the mime type id of this mime type e.g. application/json

Module contents

`blackfennec.document_system.mime_type.json` package

Submodules

`blackfennec.document_system.mime_type.json.json_mime_type` module

```
class blackfennec.document_system.mime_type.json.json_mime_type.JsonMimeType(structure_serializer:
                                                                    StructureSerial-
                                                                    izer)
```

Bases: *MimeType*

export_structure(*output*: IO, *structure*: Structure)

Exports a structure to a IO Stream

Parameters

- **output** (IO) – The IO Stream to export to
- **structure** (Structure) – The structure to export

import_structure(*to_import*: IO) → Structure

Imports a structure from a IO Stream

Parameters

to_import (IO) – The IO Stream to import from

Returns

The imported structure

Return type

Structure

property mime_type_id: str

Returns the mime type id of this mime type e.g. application/json

blackfennec.document_system.mime_type.json.json_pointer_serializer module

class

```
blackfennec.document_system.mime_type.json.json_pointer_serializer.JsonPointerSerializer
```

Bases: object

JsonPointer Implementation according to RFC6901

Class is able to serialize and deserialize JsonPointers in relative and absolute format.

```
ABSOLUTE_POINTER_PATTERN = re.compile('^/(?(([^/~])|(~[01]))*)+$')
```

```
RELATIVE_POINTER_PATTERN =
```

```
re.compile('^([0-9]+([+][0-9]+|[-][0-9]+)?)/(([^/~])|(~[01]))*$$')
```

classmethod deserialize_absolute_pointer(*json_pointer*: str) →

list[blackfennec.structure.reference_navigation.navigator.Navigator]

Parses absolute JsonPointer.

Returns

Parsed absolute json pointer

Return type

Navigator

classmethod deserialize_relative_pointer(*json_pointer*: str) →

list[blackfennec.structure.reference_navigation.navigator.Navigator]

Deserialized relative JsonPointer.

Parameters

json_pointer (str) – Json pointer in string form

Returns

Parsed json pointer

Return type

Navigator

Raises

ValueError – if the first navigator in path did not match the expected format. `[0-9]+([+-][0-9]+)?`

classmethod `is_absolute_json_pointer(pointer: str)`

Checks if pointer is absolute JsonPointer

Parameters

pointer (*str*) – JsonPointer string

Returns

True if pointer is absolute JsonPointer

Return type

bool

classmethod `is_relative_json_pointer(pointer: str)`

Checks if pointer is relative JsonPointer

Parameters

pointer (*str*) – JsonPointer string

Returns

True if pointer is relative JsonPointer

Return type

bool

classmethod `serialize(navigator_list:`

`list[blackfennec.structure.reference_navigation.navigator.Navigator]) → str`

Serializes a list of navigators into a Json Pointer String

Parameters

navigator_list (*List [Navigator]*) – A list of navigators

Returns

A Json Pointer String

Return type

str

`blackfennec.document_system.mime_type.json.json_reference_serializer` module

class `blackfennec.document_system.mime_type.json.json_reference_serializer.JsonReferenceSerializer` (*document_system.mime_type.json.json_reference_serializer.JsonReferenceSerializer*)

Bases: object

Parses Json References

`ABSOLUTE_POINTER_PATTERN = re.compile('^(/?(?([^\~])|(~[01]))*)+$')`

`REFERENCE_KEY = '$ref'`

```
RELATIVE_POINTER_PATTERN =
re.compile('^([0-9]+([+][0-9]+|[-][0-9]+)?)/(([^/~]|(~[01])))*$')

deserialize(raw: dict) → list[blackfennec.structure.reference_navigation.navigator.Navigator]

classmethod is_reference(raw: dict)

serialize(navigator_list: list[blackfennec.structure.reference_navigation.navigator.Navigator]) → dict
    Serializes a list of navigators into a json reference string

    Parameters
        navigator_list (list[Navigator]) – A list of navigators

    Returns
        A json reference string

    Return type
        str
```

Module contents

Submodules

blackfennec.document_system.mime_type.mime_type module

```
class blackfennec.document_system.mime_type.mime_type.MimeType
    Bases: object

    abstract export_structure(output: IO, structure: Structure)
        Export the structure to output IO

        Raises
            NotImplementedError – if subclass did not implement this method

    abstract import_structure(data: IO) → Structure
        Import the structure from IO data

        Returns
            Structure contained in raw data

        Return type
            Structure

        Raises
            NotImplementedError – if subclass did not implement this method

    abstract property mime_type_id: str
        Identification of mime type via ID

        Returns
            mime type ID

        Return type
            str

        Raises
            NotImplementedError – if subclass did not implement this property
```


static `try_determine_mime_type(uri: str, resource_type: ResourceType) → str`

Get mime_type through different approaches.

Tries guessing the mime_type from the URI ending, then by retrieving the content-type of the URI if the UriType is HOST_URI.

Parameters

- **uri** (*str*) – of which the mime_type should be searched
- **resource_type** (*ResourceType*) – type of the passed uri

Returns

mime_type

Return type

str

Raises

ValueError – if no mime_type could have been guessed automatically.

blackfennec.document_system.mime_type.mime_type_registry module

class `blackfennec.document_system.mime_type.mime_type_registry.MimeTypeRegistry`

Bases: object

MimeType Registry Class

Is a register of all registered mime_types.

`_mime_types`

stores internal mime_types

`deregister_mime_type(mime_type_id)`

Function to deregister a mime_type from the registry if its class matches the passed type

Parameters

mime_type_id (*str*) – key in the mime_type dict

property `mime_types`

mime_types getter

Returns

of mime_type

Return type

dict

`register_mime_type(mime_type_id: str, mime_type: MimeType)`

Function to register a new mime_type

Parameters

- **mime_type_id** (*str*) – Key at which the MimeType is to be inserted
- **mime_type** (*MimeType*) – future element of the mime_type registry

Module contents

`blackfennec.document_system.resource_type` package

Subpackages

`blackfennec.document_system.resource_type.protocols` package

Submodules

`blackfennec.document_system.resource_type.protocols.bftype_resource_type` module

`class blackfennec.document_system.resource_type.protocols.bftype_resource_type.BFTypeResourceType`(*type_re*)

Bases: *ResourceType*

`guess_mime_type`(*uri: str*)

`load_resource`(*document: Document, mode: str*)

Load the resource

Parameters

document (*Document*) – document to load

Returns

loaded resource

Return type

IO

`property protocols: List[str]`

List of protocols supported by Resource Type

Returns

protocols supported

Return type

List[str]

Raises

NotImplementedError – if subclass did not implement this property

`blackfennec.document_system.resource_type.protocols.file_resource_type` module

`class`

`blackfennec.document_system.resource_type.protocols.file_resource_type.FileResourceType`

Bases: *ResourceType*

`load_resource`(*document: Document, mode: str*) → *IO*

Load the resource

Parameters

document (*Document*) – document to load

Returns

loaded resource

Return type

IO

property protocols: `List[str]`

List of protocols supported by Resource Type

Returns

protocols supported

Return type`List[str]`**Raises****NotImplementedError** – if subclass did not implement this property**blackfennec.document_system.resource_type.protocols.https_resource_type module****class**`blackfennec.document_system.resource_type.protocols.https_resource_type.HttpsResourceType`Bases: `ResourceType`**guess_mime_type**(*uri: str*)**load_resource**(*document: Document, mode: str*) → IO

Load the resource

Parameters

- **document** (`Document`) – document to load
- **mode** (`str`) – the file open mode to use

Returns

loaded resource

Return type

IO

Raises**NotImplementedError** – if subclass did not implement this method**property protocols:** `List[str]`

List of protocols supported by Resource Type

Returns

protocols supported

Return type`List[str]`**Raises****NotImplementedError** – if subclass did not implement this property

Module contents

Submodules

blackfennec.document_system.resource_type.resource_type module

class blackfennec.document_system.resource_type.resource_type.ResourceType

Bases: object

guess_mime_type(*uri: str*) → Optional[str]

abstract load_resource(*document: Document, mode: str*) → IO

Load the resource

Parameters

- **document** (*Document*) – document to load
- **mode** (*str*) – the file open mode to use

Returns

loaded resource

Return type

IO

Raises

NotImplementedError – if subclass did not implement this method

abstract property protocols: List[str]

List of protocols supported by Resource Type

Returns

protocols supported

Return type

List[str]

Raises

NotImplementedError – if subclass did not implement this property

static try_determine_resource_type(*resource_uri: str*) → str

blackfennec.document_system.resource_type.resource_type_registry module

class

blackfennec.document_system.resource_type.resource_type_registry.ResourceTypeRegistry

Bases: object

ResourceType Registry Class

Is a register of all registered resource_types.

_resource_types

stores internal resource_types

deregister_resource_type(*resource_type_id*)

Function to deregister a `resource_type` from the registry if its class matches the passed type

Parameters

resource_type_id (*str*) – key in the `resource_type` dict

register_resource_type(*resource_type_id: str, resource_type: ResourceType*)

Function to register a new `resource_type`

Parameters

- **resource_type_id** (*str*) – Key at which the `ResourceType` is to be inserted
- **resource_type** (`ResourceType`) – future element of the `resource_type` registry

property resource_types

`resource_types` getter

Returns

of `resource_type`

Return type

dict

Module contents

Submodules

`blackfennec.document_system.document` module

class `blackfennec.document_system.document.Document`(*document_registry, mime_type: MimeType, resource_type: ResourceType, uri: str = "", location: str = ""*)

Bases: `object`

A document can contain a structure which is lazy loaded from a given URI via it's resource and mime type

property content: `Structure`

Get the content of the document

property location: `str`

Get the location of the document

property mime_type: `MimeType`

Get the mime type of the document

property resource_type: `ResourceType`

Get the resource type of the document

save()

Save the document

property uri: `str`

Get the uri of the document

blackfennec.document_system.document_factory module

```
class blackfennec.document_system.document_factory.DocumentFactory(document_registry:
    ~blackfen-
    nec.document_system.document_registry.DocumentRegistry,
    resource_type_registry:
    ~blackfen-
    nec.document_system.resource_type.resource_type_registry.ResourceTypeRegistry,
    mime_type_registry:
    ~blackfen-
    nec.document_system.mime_type.mime_type_registry.MimeTypeRegistry,
    document_type: ~typing.Type[~blackfennec.document_system.document.Document])
    = <class 'blackfennec.document_system.document_factory.DocumentFactory'>
```

Bases: object

```
create(uri: str, resource_type_id: Optional[str] = None, mime_type_id: Optional[str] = None, location:
    Optional[str] = None) → Document
```

```
get_document(structure) → Document
```

blackfennec.document_system.document_registry module

```
class blackfennec.document_system.document_registry.DocumentRegistry
```

Bases: object

```
get_document(structure: Structure) → Document
```

```
register_document(document: Document)
```

Module contents

blackfennec.extension_system package

Submodules

blackfennec.extension_system.extension module

```
class blackfennec.extension_system.extension.Extension(name: str, api: ExtensionApi, dependencies:
    Optional[set[str]] = None)
```

Bases: object

```
class State(value, names=None, *, module=None, qualname=None, type=None, start=1,
    boundary=None)
```

Bases: Enum

```
ACTIVE = 'active'
```

```
DEPENDENCY_MISSING = 'dependency_missing'
```

```

    FAILED = 'failed'
    INACTIVE = 'inactive'
activate() → None
deactivate() → None
property dependencies: set[str]
deregister_actions()
deregister_presenters()
deregister_types()
deregister_view_factories()
property is_active: bool
property name: str
register_actions()
register_presenters()
register_types()
register_view_factories()
property state: State

```

blackfennec.extension_system.extension_api module

```

class blackfennec.extension_system.extension_api.ExtensionApi(
    presenter_registry: PresenterRegistry,
    type_registry: TypeRegistry,
    interpretation_service: InterpretationService,
    view_factory: ViewFactory,
    view_factory_registry: ViewFactoryRegistry,
    type_loader: TypeLoader,
    action_registry: ActionRegistry,
    document_registry: DocumentRegistry,
    document_factory: DocumentFactory,
    ui_service: UiService,
    mime_type_registry: MimeTypeRegistry,
    resource_type_registry: ResourceTypeRegistry)

```

Bases: object

Enables extensions to load types, actions or presenters with services and registries contained within the this class.

```

property action_registry: ActionRegistry
property document_factory: DocumentFactory

```

```
property document_registry: DocumentRegistry
property interpretation_service
property mime_type_registry: MimeTypeRegistry
property presenter_registry: PresenterRegistry
property resource_type_registry: ResourceTypeRegistry
property type_loader
property type_registry: TypeRegistry
property ui_service: UiService
property view_factory
property view_factory_registry
```

blackfennec.extension_system.extension_registry module

```
class blackfennec.extension_system.extension_registry.ExtensionRegistry
    Bases: object
    get_extensions() → list[blackfennec.extension_system.extension.Extension]
    register(extension: Extension)
```

blackfennec.extension_system.extension_service module

```
class blackfennec.extension_system.extension_service.ExtensionService
    Bases: object
    classmethod load(api: ExtensionApi, registry: ExtensionRegistry)
```

Module contents

blackfennec.layers package

Subpackages

blackfennec.layers.encapsulation_base package

Submodules

blackfennec.layers.encapsulation_base.base_factory_visitor module

```
class blackfennec.layers.encapsulation_base.base_factory_visitor.BaseFactoryVisitor(layer,
                                                                                   layer_base_class)
```

Bases: *Visitor*[T]

Abstract Factory and Visitor

This class implements the base visitor behaviour

and returns the input wrapped in an adapter/decorator like manner while caching the generated generic adapter/decorator classes.

Python does not support overloading by Type, thus the visit

function are present for all existing core types.

visit_boolean(*subject*: Boolean) → T

visit_list(*subject*: List) → T

visit_map(*subject*: Map) → T

visit_null(*subject*: Null) → T

visit_number(*subject*: Number) → T

visit_reference(*subject*: Reference) → T

visit_string(*subject*: String) → T

visit_structure(*subject*: Structure) → T

blackfennec.layers.encapsulation_base.encapsulation_base module

```
class blackfennec.layers.encapsulation_base.encapsulation_base.EncapsulationBase(layer,
                                                                                   subject:
                                                                                   Structure)
```

Bases: *Structure*, *ChangeNotificationDispatchMixin*

Is the base class of the abstract visitor BaseFactoryVisitor, which means that any created object of the abstract visitor has the super class EncapsulationBase or a specialisation.

accept(*visitor*)

bind(***kwargs*)

property parent

Property for parent of this structure encapsulated in a EncapsulationBase.

property root

Property for root of this structure encapsulated in an FactoryBase.

Returns

encapsulates root of subject in FactoryBase class

Return type

EncapsulationBase

property structure

property subject

Property for access on encapsulated structure in this EncapsulationBase.

property value

Property for value of this structure.

blackfennec.layers.encapsulation_base.list_encapsulation_base module

class blackfennec.layers.encapsulation_base.list_encapsulation_base.**ListEncapsulationBase**(**kwargs)

Bases: *EncapsulationBase, List*

Base Class for encapsulations of a List.

Contains List specific overrides of certain functions

to ensure the encapsulation of any Structure returned in order to stay in the encapsulation layer.

add_item(item: Structure)

Append item to list type.

Parameters

item (Structure) – Item to append.

remove_item(item: Structure)

Remove item from List.

Parameters

item (Structure) – Item to remove.

Raises

KeyError – If the item passed is not in list and hence cannot be removed.

replace_item(old_item: Structure, new_item: Structure) → None

Replace old_item with new_item.

Parameters

- **old_item** (Structure) – Item to be replaced.
- **new_item** (Structure) – Item to replace with.

Raises

KeyError – If old_item is not in list.

property subject: *List*

Property for access on encapsulated structure in this EncapsulationBase.

property value

Property for value of this structure.

blackfennec.layers.encapsulation_base.map_encapsulation_base module

class blackfennec.layers.encapsulation_base.map_encapsulation_base.**MapEncapsulationBase**(**kwargs)

Bases: *EncapsulationBase, Map*

Base Class for Encapsulation of a Map.

add_item(key, value: Structure)

Custom set item hook, adds self as parent or raises error.

Parameters

- **key** (str) – The key for the inserted item.

- **value** (*Structure*) – The item which will be inserted.

Raises

ValueError – If the key already exists

remove_item(*key*)

Custom delete hook, resets parent for removed structure.

Parameters

key (*any*) – The key of the item to delete.

Raises

KeyError – If the item with the key to delete is not contained in map.

rename_key(*old_key: str, new_key: str*) → None

replace_item(*key: str, value: Structure*) → None

property subject: *Map*

Property for access on encapsulated structure in this EncapsulationBase.

property value

Property for value of this structure.

blackfennec.layers.encapsulation_base.reference_encapsulation_base module

class blackfennec.layers.encapsulation_base.reference_encapsulation_base.**ReferenceEncapsulationBase**(**kwargs)

Bases: *EncapsulationBase*

Reference implementation of the EncapsulationBase class.

This class is used to provide a reference implementation of the EncapsulationBase class.

resolve()

Module contents**blackfennec.layers.merge package****Submodules****blackfennec.layers.merge.merged_layer module**

class blackfennec.layers.merge.merged_layer.**MergedLayer**

Bases: object

apply(*underlay: Structure, overlay: Structure*) → *Structure*

blackfennec.layers.merge.merged_list module

class blackfennec.layers.merge.merged_list.**MergedList**(*layer*, *underlay*: Structure, *overlay*: Structure)

Bases: *MergedStructure*

property value

Property for value of this structure.

blackfennec.layers.merge.merged_map module

class blackfennec.layers.merge.merged_map.**MergedMap**(*layer*, *underlay*: Structure, *overlay*: Structure)

Bases: *MergedStructure*

property value

Property for value of this structure.

blackfennec.layers.merge.merged_null module

class blackfennec.layers.merge.merged_null.**MergedNull**(*layer*, *underlay*: Structure, *overlay*: Structure)

Bases: *MergedStructure*

accept(*visitor*: Visitor)

property value

Property for value of this structure.

blackfennec.layers.merge.merged_phantom module

class blackfennec.layers.merge.merged_phantom.**MergedPhantom**(*parent*, *twin*)

Bases: object

A mock object to allow navigating a phantom structure for merging

accept(*visitor*)

property parent

property structure

property value

blackfennec.layers.merge.merged_structure module

class blackfennec.layers.merge.merged_structure.**MergedStructure**(*layer*, *underlay*: Structure, *overlay*: Structure)

Bases: *Structure*

accept(*visitor*: Visitor)

property parent: Optional[*MergedStructure*]

Property for parent of this structure.

property root: `Optional[MergedStructure]`

Readonly property for *Root* of this structure.

property structure: `Structure`

property subject

property value

Property for value of this structure.

blackfennec.layers.merge.merger module

class `blackfennec.layers.merge.merger.BooleanMerger(layer, overlay)`

Bases: `Merger`

visit_boolean(*underlay*)

visit_null(*underlay*)

class `blackfennec.layers.merge.merger.ListMerger(layer, overlay)`

Bases: `Merger`

visit_list(*underlay*)

visit_null(*underlay*)

class `blackfennec.layers.merge.merger.MapMerger(layer, overlay)`

Bases: `Merger`

visit_map(*underlay*)

visit_null(*underlay*)

class `blackfennec.layers.merge.merger.Merger(layer, overlay)`

Bases: `Visitor[MergedStructure]`

merge(*underlay*)

abstract visit_null(*underlay*: `Null`)

visit_structure(*unused_other*: `Structure`)

class `blackfennec.layers.merge.merger.MergerFactory(layer)`

Bases: `Visitor[Merger]`

Create a Merger using the visitor pattern

create(*overlay*: `Structure`)

visit_boolean(*overlay*: `Boolean`)

visit_list(*overlay*: `List`)

visit_map(*overlay*: `Map`)

visit_null(*overlay*: `Null`)

visit_number(*overlay*: `Number`)

`visit_reference(overlay: Reference)`

`visit_string(overlay: String)`

`class blackfennec.layers.merge.merger.NullMerger(layer, overlay)`

Bases: *Merger*

`visit_list(underlay)`

`visit_map(underlay: Map)`

`visit_null(underlay)`

`visit_structure(underlay: Structure)`

`class blackfennec.layers.merge.merger.NumberMerger(layer, overlay)`

Bases: *Merger*

`visit_null(underlay)`

`visit_number(underlay)`

`class blackfennec.layers.merge.merger.StringMerger(layer, overlay)`

Bases: *Merger*

`visit_null(underlay)`

`visit_string(underlay)`

Module contents

blackfennec.layers.observable package

Submodules

blackfennec.layers.observable.observable module

`class blackfennec.layers.observable.observable.ObservableLayer`

Bases: *Observable*

`apply(structure)`

`on_changed(sender, notification: ChangeNotification)`

blackfennec.layers.observable.observable_base module

`class blackfennec.layers.observable.observable_base.ObservableBase(**kwargs)`

Bases: *EncapsulationBase*

blackfennec.layers.observable.observable_factory_visitor module

class blackfennec.layers.observable.observable_factory_visitor.**ObservableFactoryVisitor**(*layer*)
 Bases: *BaseFactoryVisitor*

Module contents**blackfennec.layers.overlay package****Submodules****blackfennec.layers.overlay.overlay module**

class blackfennec.layers.overlay.overlay.**Overlay**
 Bases: object
apply(*structure*: *Structure*) → *OverlayBase*

blackfennec.layers.overlay.overlay_base module

class blackfennec.layers.overlay.overlay_base.**OverlayBase**(*layer*, *subject*)
 Bases: *EncapsulationBase*
 Base Class for Overlay of any Structure.

blackfennec.layers.overlay.overlay_factory_visitor module

class blackfennec.layers.overlay.overlay_factory_visitor.**OverlayFactoryVisitor**(*layer*)
 Bases: *BaseFactoryVisitor*
 Overlay Factory Visitor
Class is a concrete factory which produces Overlay based
 structure encapsulations. Only few methods are overwritten which require specialised functionality. For
 all other structure types the abstract factory implementation suffices.
visit_reference(*subject_reference*: *Reference*) → *Structure*

Module contents**Module contents****blackfennec.presentation_system package****Subpackages****blackfennec.presentation_system.about_window package**

Submodules

blackfennec.presentation_system.about_window.about_window_view module

```
class blackfennec.presentation_system.about_window.about_window_view.AboutWindowView(view_model:  
    AboutWin-  
    dowView-  
    Model,  
    par-  
    ent_window)
```

Bases: object

blackfennec.presentation_system.about_window.about_window_view_model module

```
class  
blackfennec.presentation_system.about_window.about_window_view_model.AboutWindowViewModel
```

Bases: object

Module contents

blackfennec.presentation_system.extension_warning_dialog package

Submodules

blackfennec.presentation_system.extension_warning_dialog.extension_warning_dialog module

Module contents

blackfennec.presentation_system.main_window package

Submodules

blackfennec.presentation_system.main_window.black_fennec_view module

blackfennec.presentation_system.main_window.black_fennec_view_model module

```
class blackfennec.presentation_system.main_window.black_fennec_view_model.BlackFennecViewModel(services:  
    Ser-  
    vice-  
    Lo-  
    ca-  
    tor)
```

Bases: *Observable*

attach_tab(tab: DocumentTab)

can_handle_uri(uri: str) → bool

close_file(*tab*: DocumentTab)

Closes a file

Parameters

tab (DocumentTab) – tab to close

copy() → *BlackFennecViewModel*

property current_directory

detach_tab(*tab*: DocumentTab)

get_about_window_view_model()

open(*uri*: str)

open_file(*uri*: str)

Opens a file specified by the filename

Parameters

uri (str) – URI of the file to open

redo(*tab*: DocumentTab)

save(*tab*: DocumentTab)

Saves the passed file

save_all()

Saves all open files

save_as(*tab*: DocumentTab, *uri*: str)

Saves the passed tab under new path

undo(*tab*: DocumentTab)

blackfennec.presentation_system.main_window.document_tab module

```
class blackfennec.presentation_system.main_window.document_tab.DocumentTab(presenter_registry:
    PresenterRegistry,
    document_factory:
    DocumentFactory,
    navigation_service:
    NavigationService, uri: str, icon:
    str = 'emblem-
documents-
symbolic')
```

Bases: *Observable*

create_presenter()

property icon

load_document()

save_document()

`save_document_as(uri: str)`

property `uri`

`blackfennec.presentation_system.main_window.document_tab_view` module

```
class blackfennec.presentation_system.main_window.document_tab_view.DocumentTabView(tab_view:
    <Mag-
    ic-
    Mock
    name='mock.TabView'
    id='139916547025424'>
    tab:
    ~black-
    fen-
    nec.presentation_system.n
```

Bases: `object`

`blackfennec.presentation_system.main_window.file_column_view` module

Module contents

`blackfennec.presentation_system.navigation_service` package

Submodules

`blackfennec.presentation_system.navigation_service.navigation_proxy` module

```
class blackfennec.presentation_system.navigation_service.navigation_proxy.NavigationProxy
```

Bases: `Observable`

A proxy for navigation requests.

The navigation proxy dispatches requests to another interpretation.

navigate(*sender*: `Interpretation`, *destination*: `Structure`)

Navigate to destination, sender is ignored.

This function dispatches the navigation request to the configured interpretation, discarding the sender

Parameters

- **sender** (`Interpretation`) – Ignored
- **destination** (`Structure`) – destination which will be passed on

blackfennec.presentation_system.navigation_service.navigation_service module**class**`blackfennec.presentation_system.navigation_service.navigation_service.NavigationService`Bases: `object`

Navigation Service Class.

Can be called by an interpretation for navigational purposes. Class resolves route through auctioneer and dispatches navigation request to structure presenter.

`_presenter`

stores injected structure presenter

Type`StructurePresenter`**`navigate`**(*sender*: `Interpretation`, *destination*: `Structure`)

Navigation request dispatch

Dispatches navigation request received by `Interpretation` to `column_based_presenter` to show navigation with the help of the received `interpretation_service`

Parameters

- **`sender`** (`Interpretation`) – Interpretation which invoked navigation
- **`destination`** (`Structure`) – Destination to which shall be navigated

Raises

`AssertionError` – if no presenter is set for `navigation_service`. can be done with `set_presenter()`

`set_presenter`(*presenter*)

Set target for navigation requests

Parameters

`presenter` – The presenter must have the *show* method

Module contents**blackfennec.presentation_system.ui_service package****Subpackages****blackfennec.presentation_system.ui_service.mime_type_selection package****Submodules****blackfennec.presentation_system.ui_service.mime_type_selection.mime_type_selection_view module****Module contents****Submodules**

blackfennec.presentation_system.ui_service.message module

```
class blackfennec.presentation_system.ui_service.message.Message(text: str, timeout: int = None,
                                                                action_name: str = None,
                                                                action_target: str = None)
```

Bases: object

property action_name: str

property action_target: str

property text: str

property timeout: int

blackfennec.presentation_system.ui_service.ui_service module

```
class blackfennec.presentation_system.ui_service.ui_service.UiService(mime_type_registry)
```

Bases: object

```
deregister_message_overlay(message_overlay: <MagicMock name='mock.ToastOverlay'
                             id='139916545977680'>)
```

```
classmethod fix_focus(context: Context)
```

By using `Gtk.Window.set_focus()` without parameters focus is reset which can fix the focus in the window. This is a workaround for a bug only triggered in Flatpak.

```
classmethod get_clipboard_async(callback)
```

```
get_structure_from_clipboard_async(context: Context, mime_type: MimeType, callback: callable)
```

```
register_message_overlay(message_overlay: <MagicMock name='mock.ToastOverlay'
                             id='139916545977680'>)
```

```
classmethod set_clipboard(serialized_structure: str)
```

```
show_message(context: Context, message: Message)
```

```
show_mime_type_selection_dialog(context: Context, callback: callable, serialized_structure: str)
```

Module contents

Submodules

blackfennec.presentation_system.history_service module

```
class blackfennec.presentation_system.history_service.HistoryEntry(structure, old, new)
```

Bases: object

```
class blackfennec.presentation_system.history_service.HistoryService
```

Bases: object

```
can_redo()
```

can_undo()
observe(*observable*: [Observable](#))
redo()
undo()

blackfennec.presentation_system.presenter_registry module

class blackfennec.presentation_system.presenter_registry.**PresenterRegistry**

Bases: object

Presenter Registry Class

Is a register of all registered presenters.

_presenters

stores internal presenters

deregister_presenter(*presenter_type*: *type*)

Function to deregister a presenter from the registry if its class matches the passed type

Parameters

presenter_type (*type*) – element in the type list

property presenters

presenters getter

Returns

of presenter

Return type

list

register_presenter(*presenter*)

Function to register a new presenter

Parameters

presenter (*Presenter*) – future element of the presenter registry

blackfennec.presentation_system.structure_view_factory module

class blackfennec.presentation_system.structure_view_factory.**StructureViewFactory**(*type_view_factory_registry*:

[Type-ViewFactoryRegistry](#))

Bases: object

create(*interpretation*: [Interpretation](#)) → [Iterator](#)[[TypeView](#)]

blackfennec.presentation_system.type_view module

class blackfennec.presentation_system.type_view.TypeView

Bases: object

Base class for all views.

A view is a widget that represents the visualisation of a type.

blackfennec.presentation_system.type_view_factory module

class blackfennec.presentation_system.type_view_factory.TypeViewFactory

Bases: object

Base class for all view factories.

A view factory is a class that creates views for a specific type.

abstract create(*interpretation: Interpretation*) → *TypeView*

creates a view

Parameters

interpretation (*Interpretation*) – The overarching interpretation.

Returns

View

abstract satisfies(*specification: Specification*) → bool

Test if this view factory can satisfy the specification

Parameters

specification (*Specification*) – the specification to be satisfied

Returns

True if the specification can be satisfied. Otherwise False.

Return type

bool

blackfennec.presentation_system.type_view_factory_registry module

class blackfennec.presentation_system.type_view_factory_registry.TypeViewFactoryRegistry

Bases: object

View_factory Registry Class

Is a register of all known or registered view_factories.

_type_view_factories

stores internal view_factories

deregister_type_view_factory(*type: Type, specification: Specification*)

Function to deregister a type_view_factory from the dictionary if its class matches the passed type_view_factory

Parameters

- **type** (*type*) – type of type_view_factory

- **specification** (`Specification`) – specification of `type_view_factory`

get_factory(*type*, *specification*)

register_type_view_factory(*type*: `Type`, *specification*: `Specification`, *type_view_factory*: `TypeViewFactory`)

Function to register a new `type_view_factory`

Parameters

- **type** (*type*) – type of `type_view_factory`
- **specification** (`Specification`) – specification of `type_view_factory`
- **type_view_factory** (`TypeViewFactory`) – future element of the `type_view_factory` list

Module contents

blackfennec.structure package

Subpackages

blackfennec.structure.reference_navigation package

Submodules

blackfennec.structure.reference_navigation.child_navigator module

class `blackfennec.structure.reference_navigation.child_navigator.ChildNavigator`(*token*: `str`)

Bases: `Navigator`

navigate(*current*: `Union[Map, List]`) → `Structure`

navigates current structure and returns destination

Returns

Structure navigated to

Return type

`Structure`

blackfennec.structure.reference_navigation.index_of_navigator module

class `blackfennec.structure.reference_navigation.index_of_navigator.IndexOfNavigator`

Bases: `Navigator`

navigate(*current*: `Structure`) → `Structure`

navigates to index/key of structure

Returns

Structure navigated to

Return type

`Structure`

blackfennec.structure.reference_navigation.navigator module

class blackfennec.structure.reference_navigation.navigator.**Navigator**

Bases: object

abstract navigate(*current: Structure*) → *Structure*

navigates current structure and returns destination

Returns

Structure navigated to

Return type

Structure

Raises

NotImplementedError – if subclass did not implement this method

blackfennec.structure.reference_navigation.parent_navigator module

class blackfennec.structure.reference_navigation.parent_navigator.**ParentNavigator**

Bases: *Navigator*

navigate(*current: Structure*) → *Structure*

navigates current structure and returns destination

Returns

Parent

Return type

Structure

blackfennec.structure.reference_navigation.root_navigator module

class blackfennec.structure.reference_navigation.root_navigator.**RootNavigator**

Bases: *Navigator*

navigate(*current: Structure*) → *Structure*

navigates current structure and returns destination

Returns

Parent

Return type

Structure

blackfennec.structure.reference_navigation.sibling_offset_navigator module

class blackfennec.structure.reference_navigation.sibling_offset_navigator.**SiblingOffsetNavigator**(*sibling_*
int)

Bases: *Navigator*

navigate(*current: Structure*) → *Structure*

navigates current structure and returns destination

Returns

Structure navigated to

Return type

Structure

blackfennec.structure.reference_navigation.uri_navigator module

class blackfennec.structure.reference_navigation.uri_navigator.**UriNavigator**(*document_factory: DocumentFactory, uri: str*)

Bases: *Navigator*

navigate(*current: Structure*) → *Structure*

navigates current structure and returns destination

Returns

Uri content

Return type

Structure

Module contents

Submodules

blackfennec.structure.boolean module

class blackfennec.structure.boolean.**Boolean**(*value: bool = False*)

Bases: *ValueStructure*[bool]

Core Type Boolean, represents booleans in the domain model.

accept(*visitor: Visitor[TVisitor]*) → TVisitor

blackfennec.structure.list module

class blackfennec.structure.list.**List**(*value: list[T] = None*)

Bases: *Structure*[list[T]]

Core type List, a list of Structures

accept(*visitor: Visitor[TVisitor]*) → TVisitor

add_item(*item: T*) → None

Append item to list.

Parameters

item (*Structure*) – Item to append.

remove_item(*item: T*) → None

Remove item from List.

Parameters

item ([Structure](#)) – Item to remove.

Raises

KeyError – If the item passed is not in list and hence cannot be removed.

replace_item(*old_item: T, new_item: T*) → None

Replace old_item with new_item.

Parameters

- **old_item** ([Structure](#)) – Item to be replaced.
- **new_item** ([Structure](#)) – Item to replace with.

Raises

KeyError – If old_item is not in list.

property value: `list[T]`

Property for value of this structure.

blackfennec.structure.map module

class `blackfennec.structure.map.Map`(*value: dict[str, T] = None*)

Bases: [Structure](#)[dict[str, T]]

Core type Map, a set of keys with values

accept(*visitor: Visitor[TVisitor]*) → TVisitor

add_item(*key: str, value: T*) → None

Custom set item hook, adds self as parent or raises error.

Parameters

- **key** (*str*) – The key for the inserted item.
- **value** ([Structure](#)) – The item which will be inserted.

Raises

ValueError – If the key already exists

remove_item(*key: str*) → None

Custom delete hook, resets parent for removed structure.

Parameters

key (*any*) – The key of the item to delete.

Raises

KeyError – If the item with the key to delete is not contained in map.

rename_key(*old_key: str, new_key: str*) → None

replace_item(*key: str, new_value: T*) → None

property value: `dict[str, T]`

Property for value of this structure.

blackfennec.structure.null module

class blackfennec.structure.null.**Null**

Bases: *Structure*[None]

Core Type Null, represents null values in the domain model.

accept(*visitor*: *Visitor*[*TVisitor*]) → *TVisitor*

property value: **None**

Property for value of this structure.

blackfennec.structure.number module

class blackfennec.structure.number.**Number**(*value*: *int* | *float* = 0)

Bases: *ValueStructure*[*int* | *float*]

Core Type Number, represents numbers in the domain model.

accept(*visitor*: *Visitor*[*TVisitor*]) → *TVisitor*

blackfennec.structure.reference module

class blackfennec.structure.reference.**Reference**(*navigators*:
list[blackfennec.structure.reference_navigation.navigator.Navigator])

Bases: *Structure*[*list*[*Navigator*]]

Core Type Reference, represents references in the domain model.

TYPE = **None**

accept(*visitor*: *Visitor*[*TVisitor*]) → *TVisitor*

resolve() → *Structure*

Resolves Reference navigation

Returns

destination to which the reference_navigation points

Return type

Structure

property value:

list[blackfennec.structure.reference_navigation.navigator.Navigator]

Property for value of this structure.

blackfennec.structure.string module

class blackfennec.structure.string.**String**(*value: str = ""*)
Bases: *ValueStructure*[str]
Core Type String, represents strings in the domain model.
accept(*visitor: Visitor*[TVisitor]) → TVisitor

blackfennec.structure.structure module

class blackfennec.structure.structure.**Structure**
Bases: *Observable*, Generic[T]
Abstract base class for all types (Structures).
abstract accept(*visitor: Visitor*[TVisitor]) → TVisitor
property parent: Optional[*Structure*]
Property for parent of this structure.
property root: Optional[*Structure*]
Readonly property for *Root* of this structure.
property structure
abstract property value: T
Property for value of this structure.
class blackfennec.structure.structure.**ValueStructure**(*value: T*)
Bases: *Structure*[T]
Abstract base class for all structures that have a value.
property value: T
Property for value of this structure.

blackfennec.structure.structure_serializer module

class blackfennec.structure.structure_serializer.**StructureSerializer**(*reference_serializer: JsonReferenceSerializer*)
Bases: object
Serializer allows black-fennec structures to be serialized to native Python datatypes and vice versa
deserialize(*raw: Optional*[Union[dict, list, str, bool, Number]]) → *Structure*
Checks if object is an instance of a specific type and returns the parsed black-fennec structure
Parameters
raw (Union[dict, list, str, bool, numbers.Number, None]) – native Python datatype
Returns
Subclass of Structure
Return type
Structure

Raises

NotImplementedError – If the type contained in the passed native Python datatype is unhandled.

serialize(*structure*: [Structure](#)) → Optional[Union[dict, list, str, bool, Number]]

Serializes a black-fennec structure to a native Python datatype

Parameters

structure ([Structure](#)) – A black-fennec structure

Returns

A native Python datatype

Return type

Union[dict, list, str, bool, numbers.Number, None]

Raises

NotImplementedError – If the type contained in the passed Structure is unhandled

blackfennec.structure.visitor module

class `blackfennec.structure.visitor.Visitor`

Bases: `Generic[T]`

Base class for all visitors as defined by the visitor pattern.

This class is generic in T, the return type of the `visit_*` methods.

visit_boolean(*subject*: `Boolean`) → T

visit_list(*subject*: `List`) → T

visit_map(*subject*: `Map`) → T

visit_null(*subject*: `Null`)

visit_number(*subject*: `Number`) → T

visit_reference(*subject*: `Reference`) → T

visit_string(*subject*: `String`) → T

visit_structure(*subject*: `Structure`) → T

Module contents

blackfennec.type_system package

Subpackages

blackfennec.type_system.interpretation package

Submodules

blackfennec.type_system.interpretation.coverage module

```
class blackfennec.type_system.interpretation.coverage.Coverage(subject_node_count,  
                                                             type_node_count)
```

Bases: *Comparable*

Coverage calculated on a Type

Only two comparison operators are implemented(eq,lt) the rest is included via inheritance/mixin.

```
COVERED = Coverage(1, 1)
```

```
NOT_COVERED = Coverage(1, 0)
```

```
is_covered() → bool
```

blackfennec.type_system.interpretation.interpretation module

```
class blackfennec.type_system.interpretation.interpretation.Interpretation(structure:  
                                                                           Structure,  
                                                                           specification:  
                                                                           Specification,  
                                                                           types)
```

Bases: object

Interpretation Class.

Is produced by the InterpretationService. Contains the relevant structure and can create a view. Dispatches navigation requests to navigation_service.

```
navigate(destination: Structure)
```

Navigation dispatch.

Navigation request is dispatched to navigation_service. The sender is set to self.

Parameters

destination (*Structure*) – where to navigate to

```
set_navigation_service(navigation_service) → None
```

Set navigation service to be used.

Parameters

navigation_service (*NavigationService*) – The navigation service. Must not be None.

property specification: *Specification*

property structure: *Structure*

structure getter

Returns

the structure that is represented by this interpretation.

Return type

Structure

property types

blackfennec.type_system.interpretation.interpretation_service module

class blackfennec.type_system.interpretation.interpretation_service.**InterpretationService**(*type_registry: TypeRegistry*)

Bases: object

Interpretation Service Class.

Is produced during the selection process and is the Creator of Interpretations

interpret(*structure: Structure, specification: Specification*) → *Interpretation*

Interpret the given structure following the a specification

Parameters

- **structure** (*Structure*) – The structure to be interpreted
- **specification** (*Specification, optional*) – The specification to be followed. Defaults to default constructed Specification.

Returns

Represents what black fennec believes to be
the meaning of the structure.

Return type

Interpretation

blackfennec.type_system.interpretation.offer module

class blackfennec.type_system.interpretation.offer.**Offer**(*subject: Structure, type: Type*)

Bases: *Comparable*

Offer that is sent to Auctioneer by StructureViewBidder.

Only two comparison operators are implemented(eq,lt) the rest is included via inheritance/mixin.

_subject

Structure that is auctioned

Type

Structure

_specificity

Describes inheritance hierarchy level

Type

Int

_view_factory

View Factory for corresponding type

_type

structure that type can handle

_coverage

Describes coverage of nodes of subject

Type

Coverage

property coverage: *Coverage*

coverage getter

Returns

coverage property set by constructor

Return type

float

property specificity: **int**

specificity getter

Returns

specificity property set by constructor

Return type

int

property subject: *Structure*

subject getter

Returns

subject property set by constructor

Return type

Structure

property type: *Type*

type getter

Returns

Type property set by constructor

Return type

Structure

blackfennec.type_system.interpretation.specification module

```
class blackfennec.type_system.interpretation.specification.Specification(request_preview:  
                                                                    bool = False)
```

Bases: object

Specification, specifies which interpretation is desired.

This allows the manipulation of the interpreter such that the result

better suites the envisioned use case of the interpretation. For example, the callee of the interpretation service can request previews.

property is_request_for_preview: **bool**

Property to query if this specification requests a preview.

Returns

true iff a preview has been requested.

Return type
bool

Module contents

Submodules

blackfenec.type_system.boolean_type module

class blackfenec.type_system.boolean_type.**BooleanType**(*subject*: Map = None)

Bases: *Type*

Base Class for Type of a Boolean.

property default

property expected: Optional[bool]

visit_boolean(*subject*: Boolean) → *Coverage*

blackfenec.type_system.list_type module

class blackfenec.type_system.list_type.**ListType**(*subject*: Map = None)

Bases: *Type*[List]

Base Class for Type of a List.

add_element(*type*, *is_required=True*) → None

property default

property elements: list[blackfenec.type_system.type.Type]

is_child_optional(*child*: Type) → bool

property required_elements: list[blackfenec.structure.number.Number]

set_is_child_optional(*child*: Type, *is_optional*: bool) → None

set_required(*index*: int, *value*: bool) → None

visit_list(*subject*: List)

Coverage calculation for List Class

Subject may contain a type multiple times, which will be then matched by a single child of the List type multiple times.

Parameters

subject (List) – List for which coverage is calculated

Returns

of subject by self(Type)

Return type

Coverage

blackfennec.type_system.map_type module

class blackfennec.type_system.map_type.**MapType**(*subject: Map = None*)

Bases: *Type[Map]*

Base Class for Type of a Map.

add_property(*name, type: Type, is_required=True*) → None

property default

is_child_optional(*child: Type*) → bool

property properties: Dict[str, *Type*]

property required_properties: list[*blackfennec.structure.string.String*]

set_is_child_optional(*child: Type, is_optional: bool*) → None

set_required(*name: str, value*) → None

visit_map(*subject: Map*) → *Coverage*

Coverage calculation for Map Class

Parameters

subject (*Map*) – Map for which coverage is calculated

Returns

of subject by self(*Type*)

Return type

Coverage

blackfennec.type_system.null_type module

class blackfennec.type_system.null_type.**NullType**(*subject: Map = None*)

Bases: *Type*

Base Class for Type of a Null.

property default

visit_null(*subject: Null*) → *Coverage*

blackfennec.type_system.number_type module

class blackfennec.type_system.number_type.**NumberType**(*subject: Map = None*)

Bases: *Type[Number]*

Base Class for Type of a Number.

property default: *Number*

property maximum

property minimum

visit_number(*subject: Number*) → *Coverage*

blackfennec.type_system.reference_type module

class blackfennec.type_system.reference_type.**ReferenceType**(*subject*: Map = None)

Bases: *Type*

Base Class for Type of a Boolean.

property default

visit_reference(*subject*: Reference) → Coverage

blackfennec.type_system.string_type module

class blackfennec.type_system.string_type.**StringType**(*subject*: Map = None)

Bases: *Type[String]*

Base Class for Type of a String.

property default

property pattern: Pattern

validate(*subject*)

visit_string(*subject*: String) → Coverage

Check value of String for regexp

Checks whether the value contained in the type
if any can be matched with the strings value.

Parameters

subject (List) – String whose value has to match type

Returns

Coverage.COVERED if the match was successful

or no regex was contained in the type value; Coverage.NOT_COVERED if the match failed.

Return type

Coverage

blackfennec.type_system.type module

class blackfennec.type_system.type.**Type**(*subject*)

Bases: *TypeCoverageMixin*, Generic[T]

create_instance()

abstract property default: T

property is_optional

property name

property parent

property subject

property super

blackfennec.type_system.type_coverage_mixin module

class blackfennec.type_system.type_coverage_mixin.TypeCoverageMixin

Bases: *Visitor*[*Coverage*]

Base Class for Type of a any Structure.

Contains decorating additional property optional,
that can be set on a Type to indicate optionality

calculate_coverage(*subject*)

calculate the coverage of subject by this type

Parameters

subject (*Info*) – The subject which should be covered

Returns

The coverage report.

Return type

Coverage

visit_boolean(*subject*: Boolean) → *Coverage*

visit_list(*unused_arg*: List) → *Coverage*

visit_map(*unused_arg*: Map) → *Coverage*

visit_null(*unused_arg*: Null) → *Coverage*

visit_number(*subject*: Number) → *Coverage*

visit_reference(*subject*: Reference) → *Coverage*

visit_string(*unused_arg*: String) → *Coverage*

blackfennec.type_system.type_factory module

class blackfennec.type_system.type_factory.TypeFactory

Bases: object

classmethod **create_boolean**(*expected=None, default=False*)

classmethod **create_list**()

classmethod **create_map**(*properties=None, type='Map', super: Type = None*)

classmethod **create_null**()

classmethod **create_number**(*min=None, max=None, default=0*)

classmethod **create_string**(*pattern='.*', default=""*)

blackfennec.type_system.type_loader module

```
class blackfennec.type_system.type_loader.TypeLoader(document_factory, type_registry)
    Bases: object
    load(absolute_path: str)
```

blackfennec.type_system.type_parser module

```
class blackfennec.type_system.type_parser.TypeParser
    Bases: object
    FACTORIES = {'Boolean': <class 'blackfennec.type_system.boolean_type.BooleanType'>,
                 'List': <class 'blackfennec.type_system.list_type.ListType'>, 'Map': <class
                 'blackfennec.type_system.map_type.MapType'>, 'Null': <class
                 'blackfennec.type_system.null_type.NullType'>, 'Number': <class
                 'blackfennec.type_system.number_type.NumberType'>, 'Reference': <class
                 'blackfennec.type_system.reference_type.ReferenceType'>, 'String': <class
                 'blackfennec.type_system.string_type.StringType'>}
    classmethod parse(structure: Map)
```

blackfennec.type_system.type_registry module

```
class blackfennec.type_system.type_registry.TypeRegistry
    Bases: object
    Type Registry Class
    Is a register of all known or registered types.
    _types
        stores internal types
    deregister_type(type: Type)
        Function to deregister a type from the dictionary if its class
        matches the passed type
        Parameters
            type (Type) – element in the type list
    register_type(type: Type)
        Function to register a new type
        Parameters
            type_bidder (StructureBidder) – future element of the type list
    property types
        types getter
        Returns
            of type_bidder
        Return type
            list
```

Module contents

blackfennec.util package

Submodules

blackfennec.util.change_notification module

class blackfennec.util.change_notification.ChangeNotification(*old_value*, *new_value*)

Bases: object

property new_value

property old_value

blackfennec.util.change_notification_dispatch_mixin module

class blackfennec.util.change_notification_dispatch_mixin.ChangeNotificationDispatchMixin

Bases: *Observable*

Mixin class for dispatching change notifications.

blackfennec.util.comparable module

class blackfennec.util.comparable.Comparable

Bases: object

Comparable mixin

Only two comparison operators have to be implemented(*eq*,*lt*) and the rest can be included via this class.

blackfennec.util.deep_copy module

class blackfennec.util.deep_copy.DeepCopy

Bases: object

static copy(*structure*)

class blackfennec.util.deep_copy.DeepCopyVisitor

Bases: *Visitor*[*Structure*]

Creates a deep copy of a structure

visit_boolean(*subject_boolean*: Boolean) → Boolean

visit_list(*subject*: List) → List

visit_map(*subject_map*: Map) → Map

visit_null(*unused_subject*)

visit_number(*subject_number*: Number) → Number

`visit_reference(subject: Reference) → Reference`

`visit_string(subject_string: String) → String`

`visit_structure(subject_structure: Structure)`

blackfennec.util.intercepting_visitor module

`class blackfennec.util.intercepting_visitor.InterceptingVisitor(adapter, visitor)`

Bases: `Visitor`

`visit_boolean(subject: Boolean)`

`visit_list(subject: List)`

`visit_map(subject: Map)`

`visit_null(subject: Null)`

`visit_number(subject: Number)`

`visit_reference(subject: Reference)`

`visit_string(subject: String)`

`visit_structure(subject: Structure)`

blackfennec.util.meta_info module

`class blackfennec.util.meta_info.BlackFennecMetaInfo`

Bases: `object`

`get_app_id()`

`get_authors()`

`get_copy_right() → str`

`get_description()`

`get_home_page()`

`get_icon_path() → str`

`get_issue_page()`

`get_license()`

`get_name()`

`get_release_notes() → str`

`get_summary()`

`get_version()`

blackfennec.util.observable module

class blackfennec.util.observable.**Observable**

Bases: object

Observable Class

Base for Observable used in Observer Pattern.

bind(**kwargs)

blackfennec.util.parameterized_visitor module

class blackfennec.util.parameterized_visitor.**ParameterizedVisitor**(default=<function ParameterizedVisitor.<lambda>>, **kwargs)

Bases: *Visitor*

visit_boolean(subject: Boolean)

visit_list(subject: List)

visit_map(subject: Map)

visit_null(subject: Null)

visit_number(subject: Number)

visit_reference(subject: Reference)

visit_string(subject: String)

visit_structure(subject: Structure)

blackfennec.util.service_locator module

class blackfennec.util.service_locator.**ServiceLocator**

Bases: object

blackfennec.util.type_comparator module

class blackfennec.util.type_comparator.**TypeComparator**

Bases: *Visitor*

static compare(a, b)

visit_boolean(unused_subject: Boolean)

visit_list(unused_subject: List)

visit_map(unused_unused_subject: Map)

visit_null(unused_unused_subject: Null)

`visit_number(UNUSED_SUBJECT: Number)`
`visit_reference(UNUSED_SUBJECT: Reference)`
`visit_string(UNUSED_SUBJECT: String)`
`visit_structure(UNUSED_SUBJECT: Structure)`

Module contents

Module contents

blackfennec_doubles package

Subpackages

blackfennec_doubles.action_system package

Submodules

blackfennec_doubles.action_system.double_action module

`class blackfennec_doubles.action_system.double_action.ActionMock(type=None)`
Bases: object
`execute(context)`

blackfennec_doubles.action_system.double_action_registry module

`class blackfennec_doubles.action_system.double_action_registry.ActionRegistryMock(actions=None)`
Bases: object
`deregister_action(action)`
`get_actions(type)`
`register_action(action)`

blackfennec_doubles.action_system.double_context module

`class blackfennec_doubles.action_system.double_context.ContextMock(window=None)`
Bases: object

blackfennec_doubles.action_system.double_ui_context module

```
class blackfennec_doubles.action_system.double_ui_context.UiContextMock(root)
    Bases: object
    get_root()
```

Module contents

blackfennec_doubles.document_system package

Subpackages

blackfennec_doubles.document_system.mime_type package

Subpackages

blackfennec_doubles.document_system.mime_type.json package

Submodules

blackfennec_doubles.document_system.mime_type.json.double_json_pointer_parser module

```
class blackfennec_doubles.document_system.mime_type.json.double_json_pointer_parser.JsonPointerSerializer
```

Bases: *JsonPointerSerializer*

deserialize_absolute_pointer(*pointer: str*) → *Navigator*

Parses absolute JsonPointer.

Returns

Parsed absolute json pointer

Return type

Navigator

deserialize_relative_pointer(*pointer: str*) → *Navigator*

Deserialized relative JsonPointer.

Parameters

json_pointer (*str*) – Json pointer in string form

Returns

Parsed json pointer

Return type*Navigator***Raises****ValueError** – if the first navigator in path did not match the expected format. `[0-9]+([+][0-9]+)?`**serialize**(*navigator*: *Navigator*)

Serializes a list of navigators into a Json Pointer String

Parameters**navigator_list** (*List* [*Navigator*]) – A list of navigators**Returns**

A Json Pointer String

Return type

str

blackfennec_doubles.document_system.mime_type.json.double_json_reference_serializer module**class** blackfennec_doubles.document_system.mime_type.json.double_json_reference_serializer.**JsonReference**

Bases: object

REFERENCE_KEY = '\$ref'**deserialize**(*raw*)**classmethod** **is_reference**(*raw*: *dict*)**serialize**(*structure*)**blackfennec_doubles.document_system.mime_type.json.double_structure_serializer module****class** blackfennec_doubles.document_system.mime_type.json.double_structure_serializer.**StructureSerialize**

Bases: object

deserialize(*raw*)**serialize**(*structure*)

Module contents

Submodules

blackfennec_doubles.document_system.mime_type.double_mime_type module

```
class blackfennec_doubles.document_system.mime_type.double_mime_type.MimeTypeMock(mime_type_id='mime_type',  
                                                                                   im-  
                                                                                   ported_structure=None,  
                                                                                   ex-  
                                                                                   ported_structure=None)
```

Bases: *MimeType*

export_structure(*output: IO, structure*)

Export the structure to output IO

Raises

NotImplementedError – if subclass did not implement this method

import_structure(*data: IO*) → *StructureMock*

Import the structure from IO data

Returns

Structure contained in raw data

Return type

Structure

Raises

NotImplementedError – if subclass did not implement this method

property mime_type_id: str

Identification of mime type via ID

Returns

mime type ID

Return type

str

Raises

NotImplementedError – if subclass did not implement this property

blackfennec_doubles.document_system.mime_type.double_mime_type_registry module

```
class blackfennec_doubles.document_system.mime_type.double_mime_type_registry.MimeTypeRegistryMock(mime
```

Bases: object

deregister_mime_type(*mime_type_key*)

property mime_types

register_mime_type(*mime_key, mime_type*)

Module contents

`blackfennec_doubles.document_system.resource_type` package

Submodules

`blackfennec_doubles.document_system.resource_type.double_resource_type` module

`class blackfennec_doubles.document_system.resource_type.double_resource_type.ResourceTypeMock` (*protocols=...*, *loaded_reso...*)

Bases: *ResourceType*

`load_resource` (*document: Document, mode: str*) → IO

Load the resource

Parameters

- **document** (*Document*) – document to load
- **mode** (*str*) – the file open mode to use

Returns

loaded resource

Return type

IO

Raises

NotImplementedError – if subclass did not implement this method

`property protocols: List[str]`

List of protocols supported by Resource Type

Returns

protocols supported

Return type

List[str]

Raises

NotImplementedError – if subclass did not implement this property

`blackfennec_doubles.document_system.resource_type.double_resource_type_registry` module

`class blackfennec_doubles.document_system.resource_type.double_resource_type_registry.ResourceTypeRegis...`

Bases: object

`deregister_resource_type` (*resource_type_key*)

`register_resource_type` (*resource_key, resource_type*)

`property resource_types`

Module contents

Submodules

blackfennec_doubles.document_system.double_document module

```
class blackfennec_doubles.document_system.double_document.DocumentMock(document_registry=None, mime_type=None, resource_type=None, uri: str = None, location: str = None, content=None)
```

Bases: object

property content

save()

blackfennec_doubles.document_system.double_document_factory module

```
class blackfennec_doubles.document_system.double_document_factory.DocumentFactoryMock(mime_type_registry=None, resource_type_registry=None, document_registry=None, mime_type=None, resource_type=None, location=None)
```

Bases: object

create(*uri: str, resource_type: str = None, mime_type: str = None, location: str = None*)

get_document(*structure*)

blackfennec_doubles.document_system.double_document_registry module

```
class blackfennec_doubles.document_system.double_document_registry.DocumentRegistryMock
```

Bases: object

get_document(*content*)

register_document(*document*)

Module contents

blackfennec_doubles.extension_system package

Subpackages

blackfennec_doubles.extension_system.double_extensions package

Subpackages

`blackfennec_doubles.extension_system.double_extensions.create_failing_extension` package

Submodules

`blackfennec_doubles.extension_system.double_extensions.create_failing_extension.double_create_failing_extension` module

```
class blackfennec_doubles.extension_system.double_extensions.create_failing_extension.  
double_create_failing_extension.CreateFailingExtensionMock
```

Bases: object

```
create_extension(extension_api)
```

```
destroy_extension(extension_api)
```

Module contents

`blackfennec_doubles.extension_system.double_extensions.destroy_failing_extension` package

Submodules

`blackfennec_doubles.extension_system.double_extensions.destroy_failing_extension.double_destroy_failing_extension` module

```
class blackfennec_doubles.extension_system.double_extensions.destroy_failing_extension.  
double_destroy_failing_extension.DestroyFailingExtensionMock
```

Bases: object

```
create_extension(extension_api)
```

```
destroy_extension(extension_api)
```

Module contents

`blackfennec_doubles.extension_system.double_extensions.valid_extension` package

Submodules

`blackfennec_doubles.extension_system.double_extensions.valid_extension.double_valid_extension` module

```
class blackfennec_doubles.extension_system.double_extensions.valid_extension.  
double_valid_extension.ValidExtensionMock
```

Bases: object

```
create_extension(extension_api)
```

```
destroy_extension(extension_api)
```

Module contents

Module contents

Submodules

blackfennec_doubles.extension_system.double_extension module

```
class blackfennec_doubles.extension_system.double_extension.ExtensionMock(name, dependencies=None, expected_state=State.ACTIVE)
```

Bases: object

activate()

assert_state()

blackfennec_doubles.extension_system.double_extension_api module

```
class blackfennec_doubles.extension_system.double_extension_api.ExtensionApiMock(presenter_registry=None, interpretation_service=None, document_factory=None, document_registry=None, ui_service=None, mime_type_registry=None)
```

Bases: object

blackfennec_doubles.extension_system.double_extension_loading_service module

```
class blackfennec_doubles.extension_system.double_extension_loading_service.ExtensionLoadingServiceMock
```

Bases: object

installed(sender, name, location)

load(extension)

blackfennec_doubles.extension_system.double_extension_registry module**class**blackfennec_doubles.extension_system.double_extension_registry.**ExtensionRegistryMock**

Bases: object

get_extensions() → list**register**(*extension*)**blackfennec_doubles.extension_system.double_extension_source module****class** blackfennec_doubles.extension_system.double_extension_source.**ExtensionSourceMock**(*extensions=None*)

Bases: object

refresh_extensions()**blackfennec_doubles.extension_system.double_extension_source_registry module****class** blackfennec_doubles.extension_system.double_extension_source_registry.**ExtensionSourceRegistryMock**

Bases: object

deregister_extension_source(*extension_source*)**property extension_sources****register_extension_source**(*extension_source*)**blackfennec_doubles.extension_system.extension_api_factory module**blackfennec_doubles.extension_system.extension_api_factory.**create_extension_api**(***kwargs*)**Module contents****blackfennec_doubles.layers package****Subpackages****blackfennec_doubles.layers.encapsulation_base package****Submodules****blackfennec_doubles.layers.encapsulation_base.double_factory_base_visitor module****class** blackfennec_doubles.layers.encapsulation_base.double_factory_base_visitor.**FactoryBaseVisitorMock**(

Bases: object

get_stats(*subject_type_name*)

property `metadata_storage`
visit_boolean(*subject_boolean*)
visit_list(*subject_list*)
visit_map(*subject_map*)
visit_null(*subject*)
visit_number(*subject_number*)
visit_reference(*subject_reference*)
visit_string(*subject_string*)

Module contents

blackfennec_doubles.layers.observable package

Submodules

blackfennec_doubles.layers.observable.double_observable module

class `blackfennec_doubles.layers.observable.double_observable.ObservableLayerMock`
Bases: `object`
on_changed(*sender, notification*)

Module contents

Submodules

blackfennec_doubles.layers.double_layer module

class `blackfennec_doubles.layers.double_layer.LayerMock`(*returns=None*)
Bases: `object`
apply(*structure*)
get_stats(*subject_type_name*)

Module contents

blackfennec_doubles.presentation_system package

Subpackages

blackfennec_doubles.presentation_system.main_window package

Submodules

blackfennec_doubles.presentation_system.main_window.double_document_tab module

class blackfennec_doubles.presentation_system.main_window.double_document_tab.**DocumentTabMock**(*document*, *history=None*)

Bases: object

save_document()

save_document_as(*uri*)

Module contents

blackfennec_doubles.presentation_system.navigation_service package

Submodules

blackfennec_doubles.presentation_system.navigation_service.double_navigation_service module

NavigationService Doubles.

In this module any Doubles used for testing components using the NavigationService are contained to ensure that unit-tests only test a single component.

class blackfennec_doubles.presentation_system.navigation_service.double_navigation_service.**NavigationServiceMock**

Bases: object

Class represents a Mock Double for the NavigationService.

With this class it can be tested whether the navigate member function was called in the correct way, and how many times it was executed.

navigate(*sender*, *destination*)

StructureViewFactory.create method mock.

Parameters

- **sender** (*Any*) – sender interpretation which initiated navigation
- **destination** (*Any*) – structure to navigate to

Saves passed argument on class to enable user to see passed arguments. Counts amount of times navigate has been called.

set_presenter(*presenter*)

NavigationService.set_presenter method mock.

Parameters

- **presenter** (*Any*) – presenter to set

Saves passed argument on class on property set_presenter_param to enable user to see passed arguments. Counts amount of times set_presenter has been called.

Module contents

blackfennec_doubles.presentation_system.ui_service package

Submodules

blackfennec_doubles.presentation_system.ui_service.double_message module

```
class blackfennec_doubles.presentation_system.ui_service.double_message.MessageMock(text,  
                                            ac-  
                                            tion_name=None,  
                                            ac-  
                                            tion_target=None)
```

Bases: object

blackfennec_doubles.presentation_system.ui_service.double_message_overlay module

```
class blackfennec_doubles.presentation_system.ui_service.double_message_overlay.MessageOverlayMock(root=
```

Bases: object

```
add_toast(toast)
```

```
get_root()
```

blackfennec_doubles.presentation_system.ui_service.double_ui_service module

```
class blackfennec_doubles.presentation_system.ui_service.double_ui_service.UiServiceMock(clipboard_content=
```

Bases: object

```
copy()
```

```
fix_focus(context)
```

```
get_structure_from_clipboard_async(context, mime_type, callback)
```

```
set_clipboard(text)
```

```
show_message(context, message)
```

blackfennec_doubles.presentation_system.ui_service.double_ui_service_registry module

```
class blackfennec_doubles.presentation_system.ui_service.double_ui_service_registry.UiServiceRegistryMock
```

Bases: object

```
register(window, service)
```

```
property services
```

Module contents

Submodules

blackfennec_doubles.presentation_system.double_history_service module

class blackfennec_doubles.presentation_system.double_history_service.**HistoryServiceMock**

Bases: object

add_to_history(*entry*)

can_redo()

can_undo()

observe(*unused_observable*)

redo()

undo()

blackfennec_doubles.presentation_system.double_presenter_factory module

class

blackfennec_doubles.presentation_system.double_presenter_factory.**PresenterFactoryMock**

Bases: object

create(*navigation_service, history*)

blackfennec_doubles.presentation_system.double_presenter_registry module

class blackfennec_doubles.presentation_system.double_presenter_registry.**PresenterRegistryMock**(*presenters=*

Bases: object

deregister_presenter(*factory_type*)

property presenters

register_presenter(*factory*)

blackfennec_doubles.presentation_system.double_presenter_view module

class blackfennec_doubles.presentation_system.double_presenter_view.**PresenterViewMock**

Bases: object

blackfennec_doubles.presentation_system.double_presenter_view_model module

```
class blackfennec_doubles.presentation_system.double_presenter_view_model.PresenterViewModelMock(interpretation)
    Bases: Observable
    set_structure(structure)
    show(sender, destination)
```

blackfennec_doubles.presentation_system.double_structure_presenter module

```
class
blackfennec_doubles.presentation_system.double_structure_presenter.StructurePresenterMock
    Bases: object
    show(sender, destination)
```

blackfennec_doubles.presentation_system.double_structure_view module

blackfennec_doubles.presentation_system.double_structure_view_factory module

```
class blackfennec_doubles.presentation_system.double_structure_view_factory.StructureViewFactoryMock(view)
    Bases: object
    create(interpretation)
```

blackfennec_doubles.presentation_system.double_view_factory module

```
class blackfennec_doubles.presentation_system.double_view_factory.ViewFactoryMock(view=None)
    Bases: object
    create(interpretation)
```

blackfennec_doubles.presentation_system.double_view_factory_registry module

```
class blackfennec_doubles.presentation_system.double_view_factory_registry.ViewFactoryRegistryMock(factory)
    Bases: object
    deregister_type_view_factory(type, specification)
    get_factory(type, specification)
    register_type_view_factory(type, specification, factory)
```

Module contents

`blackfennec_doubles.structure` package

Subpackages

`blackfennec_doubles.structure.reference_navigation` package

Submodules

`blackfennec_doubles.structure.reference_navigation.double_navigation` module

`class blackfennec_doubles.structure.reference_navigation.double_navigation.NavigatorMock`(*navigate_return=None*)

Bases: `object`

`navigate`(*current_structure: Structure*) → *Structure*

Module contents

Submodules

`blackfennec_doubles.structure.double_boolean` module

`class blackfennec_doubles.structure.double_boolean.BooleanMock`(*value=None, parent=None, root=None*)

Bases: `StructureMock`

`accept`(*visitor*)

`blackfennec_doubles.structure.double_list` module

`class blackfennec_doubles.structure.double_list.ListInstanceMock`(*value=None, parent=None, root=None*)

Bases: `List, ListMock`

`class blackfennec_doubles.structure.double_list.ListMock`(*value=None, parent=None, root=None*)

Bases: `StructureMock`

`accept`(*visitor*)

`add_item`(*item*)

`remove_item`(*item*)

blackfennec_doubles.structure.double_map module

class blackfennec_doubles.structure.double_map.**MapInstanceMock**(value: dict = None, parent=None, root=None)

Bases: *Map, MapMock*

class blackfennec_doubles.structure.double_map.**MapMock**(value: dict = None, parent=None, root=None)

Bases: *StructureMock*

accept(visitor)

add_item(key, item)

remove_item(key)

blackfennec_doubles.structure.double_null module

class blackfennec_doubles.structure.double_null.**NullMock**(parent=None, root=None)

Bases: *StructureMock*

accept(visitor)

blackfennec_doubles.structure.double_number module

class blackfennec_doubles.structure.double_number.**NumberMock**(value=None, parent=None, root=None)

Bases: *StructureMock*

accept(visitor)

blackfennec_doubles.structure.double_reference module

class blackfennec_doubles.structure.double_reference.**ReferenceInstanceMock**(reference=None)

Bases: *Reference, ReferenceMock*

class blackfennec_doubles.structure.double_reference.**ReferenceMock**(value=None, parent=None, root=None, resolve_return=None)

Bases: *StructureMock*

accept(visitor)

resolve()

blackfennec_doubles.structure.double_root module

class blackfennec_doubles.structure.double_root.**RootMock**(*value=None, document=None*)

Bases: *StructureInstanceMock*

accept(*visitor*)

get_document()

property root

Readonly property for *Root* of this structure.

blackfennec_doubles.structure.double_string module

class blackfennec_doubles.structure.double_string.**StringMock**(*value=None, parent=None, root=None*)

Bases: *StructureMock*

accept(*visitor*)

blackfennec_doubles.structure.double_structure module

class blackfennec_doubles.structure.double_structure.**NotifyingStructureMock**(**args, **kwargs*)

Bases: *StructureMock*

property value

class blackfennec_doubles.structure.double_structure.**StructureInstanceMock**(*value=None, parent=None, root=None*)

Bases: *StructureMock, Structure*

accept(*visitor*)

class blackfennec_doubles.structure.double_structure.**StructureMock**(*value=None, parent=None, root=None, accept_strategy=None*)

Bases: *Observable*

accept(*visitor*)

property parent

property root

property structure

property value

Module contents

`blackfennec_doubles.type_system` package

Subpackages

`blackfennec_doubles.type_system.interpretation` package

Submodules

`blackfennec_doubles.type_system.interpretation.double_auctioneer` module

```
class blackfennec_doubles.type_system.interpretation.double_auctioneer.AuctioneerMock(factories=None)
    Bases: object
    auction(subject, specification)
```

`blackfennec_doubles.type_system.interpretation.double_coverage` module

```
class blackfennec_doubles.type_system.interpretation.double_coverage.CoverageMock(ratio)
    Bases: object
    is_covered() → bool
```

`blackfennec_doubles.type_system.interpretation.double_interpretation` module

```
class blackfennec_doubles.type_system.interpretation.double_interpretation.InterpretationMock(structure=None,
structure_view=None,
specification=None,
types=None)
    Bases: object
    navigate(structure)
    set_navigation_service(navigation_service)
    property structure
```

blackfennec_doubles.type_system.interpretation.double_interpretation_service module

```
class blackfennec_doubles.type_system.interpretation.double_interpretation_service.InterpretationService
    Bases: object
    interpret(structure, specification=None)
```

blackfennec_doubles.type_system.interpretation.double_offer module

```
class blackfennec_doubles.type_system.interpretation.double_offer.OfferFake(coverage=<blackfennec_doubles.type_system.interpretation.double_offer.Coverage object>,
    view_factory=ViewFactory,
    satisfies=True)

    Bases: object
    property coverage
    satisfies(unused_specification)
    property view_factory
```

blackfennec_doubles.type_system.interpretation.double_specification module

```
class blackfennec_doubles.type_system.interpretation.double_specification.SpecificationMock(request_preview: bool,
    is_request_for_preview: bool = False)

    Bases: object
    property is_request_for_preview: bool
```

Module contents**Submodules****blackfennec_doubles.type_system.double_type** module

```
class blackfennec_doubles.type_system.double_type.TypeMock(type=None, coverage=None,
    default=None, super_type=None)
```

Bases: *Type*

```
calculate_coverage(subject)
    calculate the coverage of subject by this type
```

Parameters

subject (*Info*) – The subject which should be covered

Returns

The coverage report.

Return type

Coverage

property default

blackfennec_doubles.type_system.double_type_loader module

class blackfennec_doubles.type_system.double_type_loader.TypeLoaderMock(*type_registry*)

Bases: object

load(*uri*)

blackfennec_doubles.type_system.double_type_parser module

class blackfennec_doubles.type_system.double_type_parser.TypeParserMock

Bases: *FactoryBaseVisitorMock*

property metadata_storage

visit_boolean(*subject_boolean*)

visit_list(*subject_list*)

visit_map(*subject_map*)

visit_null(*subject*)

visit_number(*subject_number*)

visit_reference(*subject_reference*)

visit_root(*subject_root*)

visit_string(*subject_string*)

visit_structure(*subject_structure*)

blackfennec_doubles.type_system.double_type_registry module

class blackfennec_doubles.type_system.double_type_registry.TypeRegistryMock(*types=None*)

Bases: object

deregister_type(*type*)

register_type(*type*)

property types

Module contents

blackfennec_doubles.util package

Submodules

blackfennec_doubles.util.double_change_notification module

```
class blackfennec_doubles.util.double_change_notification.ChangeNotificationMock(old_value,
                                                                              new_value)
```

Bases: object

blackfennec_doubles.util.double_comparable module

```
class blackfennec_doubles.util.double_comparable.ComparableMock(value)
```

Bases: *Comparable*

blackfennec_doubles.util.double_meta_info module

```
class blackfennec_doubles.util.double_meta_info.MetaInfoMock(version='0.0.0',
                                                             description='description',
                                                             name='BlackFennec',
                                                             summary='summary',
                                                             license='GPL-3', url='url')
```

Bases: object

`get_authors()`

`get_copy_right()` → str

`get_description()`

`get_home_page()`

`get_icon_path()`

`get_id()`

`get_issue_page()`

`get_license()`

`get_name()`

`get_summary()`

`get_version()`

blackfennec_doubles.util.double_observable module

class blackfennec_doubles.util.double_observable.**ObservableMock**

Bases: *Observable*

property i_notify_observers

blackfennec_doubles.util.double_service_locator module

class blackfennec_doubles.util.double_service_locator.**ServiceLocatorMock**

Bases: object

Module contents

Submodules

blackfennec_doubles.double_dummy module

class blackfennec_doubles.double_dummy.**Dummy**(name='dummy')

Bases: object

Class represents a generic Dummy Double.

This Class can be used as generic dummy.

Module contents

tests package

Subpackages

tests.blackfennec package

Subpackages

tests.blackfennec.action_system package

Submodules

tests.blackfennec.action_system.test_action module

class tests.blackfennec.action_system.test_action.**ConcreteAction**(type=None)

Bases: *Action*

property description

An informative description of the action.

execute(*context*)

Function to execute the action

Parameters

context (*Context*) – context of the action

property name

A short, identifying name for the action.

tests.blackfennec.action_system.test_action.test_can_construct_action()

tests.blackfennec.action_system.test_action.test_can_execute()

Ensure the interface of the execute function is correct

tests.blackfennec.action_system.test_action.test_can_get_type()

tests.blackfennec.action_system.test_action.test_must_override_execute()

tests.blackfennec.action_system.test_action_registry module

tests.blackfennec.action_system.test_action_registry.test_can_construct()

tests.blackfennec.action_system.test_action_registry.test_can_deregister_action()

tests.blackfennec.action_system.test_action_registry.test_can_get_actions()

tests.blackfennec.action_system.test_action_registry.test_can_register_action()

tests.blackfennec.action_system.test_action_registry.test_can_register_two_actions_with_same_type()

tests.blackfennec.action_system.test_action_registry.test_cannot_deregister_action_twice()

tests.blackfennec.action_system.test_action_registry.test_only_deregisters_correct_action()

tests.blackfennec.action_system.test_action_registry.test_returns_empty_list_by_default()

tests.blackfennec.action_system.test_context module

tests.blackfennec.action_system.test_context.test_can_get_structure()

Module contents

tests.blackfennec.document_system package

Subpackages

tests.blackfennec.document_system.mime_type package

Subpackages

tests.blackfennec.document_system.mime_type.json package

Submodules

tests.blackfennec.document_system.mime_type.json.test_json_mime_type module

tests.blackfennec.document_system.mime_type.json.test_json_mime_type.**json_file**(*json_string*)

tests.blackfennec.document_system.mime_type.json.test_json_mime_type.**json_mime_type**(*structure_serializer*)

tests.blackfennec.document_system.mime_type.json.test_json_mime_type.**json_string**()

tests.blackfennec.document_system.mime_type.json.test_json_mime_type.**raw_json**()

tests.blackfennec.document_system.mime_type.json.test_json_mime_type.**structure_serializer**(*raw_json*)

tests.blackfennec.document_system.mime_type.json.test_json_mime_type.**test_can_construct**(*json_mime_type*)

tests.blackfennec.document_system.mime_type.json.test_json_mime_type.**test_export_structure**(*json_mime_type*,
structure_serializer,
json_string)

tests.blackfennec.document_system.mime_type.json.test_json_mime_type.**test_import_structure**(*json_mime_type*,
structure_serializer,
json_file)

tests.blackfennec.document_system.mime_type.json.test_json_pointer_serializer module

tests.blackfennec.document_system.mime_type.json.test_json_pointer_serializer.**test_is_absolute_json_pointer**

tests.blackfennec.document_system.mime_type.json.test_json_pointer_serializer.**test_is_relative_json_pointer**

tests.blackfennec.document_system.mime_type.json.test_json_pointer_serializer.**test_parse_absolute_pointer**

tests.blackfennec.document_system.mime_type.json.test_json_pointer_serializer.**test_parse_relative_pointer**

tests.blackfennec.document_system.mime_type.json.test_json_pointer_serializer.test_serialize_relative_p

tests.blackfennec.document_system.mime_type.json.test_json_reference_serializer module

tests.blackfennec.document_system.mime_type.json.test_json_reference_serializer.document()
tests.blackfennec.document_system.mime_type.json.test_json_reference_serializer.document_factory(*document*)
tests.blackfennec.document_system.mime_type.json.test_json_reference_serializer.json_reference_parser(*d*)
tests.blackfennec.document_system.mime_type.json.test_json_reference_serializer.test_can_construct(*json_*)
tests.blackfennec.document_system.mime_type.json.test_json_reference_serializer.test_is_json_reference()
tests.blackfennec.document_system.mime_type.json.test_json_reference_serializer.test_is_json_reference_
tests.blackfennec.document_system.mime_type.json.test_json_reference_serializer.test_parse_json_referen

tests.blackfennec.document_system.mime_type.json.test_json_reference_serializer.test_serialize_json_ref

tests.blackfennec.document_system.mime_type.json.test_structure_serializer module

tests.blackfennec.document_system.mime_type.json.test_structure_serializer.json_reference_serializer()
tests.blackfennec.document_system.mime_type.json.test_structure_serializer.structure_serializer(*json_refer*)
tests.blackfennec.document_system.mime_type.json.test_structure_serializer.test_can_construct()
tests.blackfennec.document_system.mime_type.json.test_structure_serializer.test_deserialize(*raw,*
ex-
pected,
struc-
ture_serializer)
tests.blackfennec.document_system.mime_type.json.test_structure_serializer.test_serialize(*structure,*
ex-
pected,
struc-
ture_serializer)

tests.blackfennec.document_system.mime_type.json.test_structure_serializer.test_throws_error_on_deseriali

tests.blackfennec.document_system.mime_type.json.test_structure_serializer.test_throws_error_on_seriali

Module contents

Submodules

tests.blackfennec.document_system.mime_type.test_mime_type module

class tests.blackfennec.document_system.mime_type.test_mime_type.**MimeTypeTestSuite**(*methodName='runTest'*)

Bases: TestCase

setUp()

Hook method for setting up the test fixture before exercising it.

test_determine_mimetype_of_file()

test_determine_mimetype_of_unknown()

tests.blackfennec.document_system.mime_type.test_mime_type_registry module

class tests.blackfennec.document_system.mime_type.test_mime_type_registry.**MimeTypeRegistryTestSuite**(*meth*)

Bases: TestCase

test_create_mime_type_registry()

test_deregister_mime_type()

test_register_mime_type()

Module contents

tests.blackfennec.document_system.resource_type package

Subpackages

tests.blackfennec.document_system.resource_type.protocols package

Submodules

tests.blackfennec.document_system.resource_type.protocols.test_file_resource_type module

class tests.blackfennec.document_system.resource_type.protocols.test_file_resource_type.**FileResourceType**

Bases: TestCase

setUp() → None

Hook method for setting up the test fixture before exercising it.

`tearDown()` → None

Hook method for deconstructing the test fixture after testing it.

`test_load_resource_absolute_path()`

`test_load_resource_relative_path()`

`tests.blackfennec.document_system.resource_type.protocols.test_integration_https_resource_type` module

`class tests.blackfennec.document_system.resource_type.protocols.test_integration_https_resource_type.Ht`

Bases: `TestCase`

`setUp()` → None

Hook method for setting up the test fixture before exercising it.

`test_determine_mimetype_online()`

`test_load_resource()`

Module contents

Submodules

`tests.blackfennec.document_system.resource_type.test_resource_type` module

`class tests.blackfennec.document_system.resource_type.test_resource_type.ResourceTypeTestSuite` (*methodNam*

Bases: `TestCase`

`test_determine_resource_type_of_file()`

`test_determine_resource_type_of_http()`

`test_determine_resource_type_of_https()`

`tests.blackfennec.document_system.resource_type.test_resource_type_registry` module

`class tests.blackfennec.document_system.resource_type.test_resource_type_registry.ResourceTypeRegistryT`

Bases: `TestCase`

`test_create_resource_type_registry()`

`test_deregister_resource_type()`

`test_register_resource_type()`

Module contents

Submodules

tests.blackfennec.document_system.test_document module

class tests.blackfennec.document_system.test_document.TestDocument

Bases: object

content()

document(*document_registry*, *mime_type*, *resource_type*)

document_registry()

mime_type(*content*)

resource_type()

test_can_construct(*document*)

test_does_register_document(*document*, *document_registry*)

test_get_content(*content*, *document*)

test_get_content_cached(*document*, *mime_type*, *resource_type*)

test_save(*document*, *mime_type*, *resource_type*)

tests.blackfennec.document_system.test_document_factory module

tests.blackfennec.document_system.test_document_factory.**document_factory**(*document_registry*,
re-
source_type_registry,
mime_type_registry)

tests.blackfennec.document_system.test_document_factory.**document_registry**()

tests.blackfennec.document_system.test_document_factory.**mime_type_registry**()

tests.blackfennec.document_system.test_document_factory.**resource_type_registry**()

tests.blackfennec.document_system.test_document_factory.**test_can_construct**(*document_factory*)

tests.blackfennec.document_system.test_document_factory.**test_create_document**(*document_factory*,
mime_type_registry,
re-
source_type_registry)

tests.blackfennec.document_system.test_document_factory.**test_create_document_without_mime_type**(*document_*
mime_type
re-
source_type)

tests.blackfennec.document_system.test_document_factory.test_get_document_for_created_document(*document_u-ment_regis*)

tests.blackfennec.document_system.test_document_registry module

tests.blackfennec.document_system.test_document_registry.test_create_document_registry()

tests.blackfennec.document_system.test_document_registry.test_get_document()

tests.blackfennec.document_system.test_document_registry.test_register_document()

Module contents

tests.blackfennec.extension_system package

Submodules

tests.blackfennec.extension_system.test_extension module

tests.blackfennec.extension_system.test_extension.extension()

tests.blackfennec.extension_system.test_extension.test_can_activate(*extension*)

tests.blackfennec.extension_system.test_extension.test_can_construct_extension(*extension*)

tests.blackfennec.extension_system.test_extension.test_can_deactivate(*extension*)

tests.blackfennec.extension_system.test_extension.test_can_get_dependencies(*extension*)

tests.blackfennec.extension_system.test_extension.test_can_get_is_active(*extension*)

tests.blackfennec.extension_system.test_extension.test_can_get_name(*extension*)

tests.blackfennec.extension_system.test_extension_api module

tests.blackfennec.extension_system.test_extension_api.action_registry()

tests.blackfennec.extension_system.test_extension_api.document_factory()

tests.blackfennec.extension_system.test_extension_api.document_registry()

```
tests.blackfennec.extension_system.test_extension_api.extension_api(presenter_registry,  
type_registry,  
interpretation_service,  
view_factory,  
view_factory_registry,  
type_loader,  
action_registry,  
document_registry,  
document_factory,  
ui_service,  
mime_type_registry,  
resource_type_registry)  
  
tests.blackfennec.extension_system.test_extension_api.interpretation_service()  
tests.blackfennec.extension_system.test_extension_api.mime_type_registry()  
tests.blackfennec.extension_system.test_extension_api.presenter_registry()  
tests.blackfennec.extension_system.test_extension_api.resource_type_registry()  
tests.blackfennec.extension_system.test_extension_api.test_can_construct(extension_api)  
tests.blackfennec.extension_system.test_extension_api.test_can_get_action_registry(extension_api,  
ac-  
tion_registry)  
tests.blackfennec.extension_system.test_extension_api.test_can_get_document_factory(extension_api,  
docu-  
ment_factory)  
tests.blackfennec.extension_system.test_extension_api.test_can_get_document_registry(extension_api,  
docu-  
ment_registry)  
tests.blackfennec.extension_system.test_extension_api.test_can_get_interpretation_service(extension_api,  
in-  
ter-  
pre-  
ta-  
tion_service)  
tests.blackfennec.extension_system.test_extension_api.test_can_get_mime_type_registry(extension_api,  
mime_type_registry)  
tests.blackfennec.extension_system.test_extension_api.test_can_get_presenter_registry(extension_api,  
pre-  
sen-  
ter_registry)  
tests.blackfennec.extension_system.test_extension_api.test_can_get_resource_type_registry(extension_api,  
re-  
source_type_regi)  
tests.blackfennec.extension_system.test_extension_api.test_can_get_type_loader(extension_api,  
type_loader)
```

```

tests.blackfennec.extension_system.test_extension_api.test_can_get_type_registry(extension_api,
                                                                    type_registry)
tests.blackfennec.extension_system.test_extension_api.test_can_get_ui_service_registry(extension_api,
                                                                    ui_service)
tests.blackfennec.extension_system.test_extension_api.test_can_get_view_factory(extension_api,
                                                                    view_factory)
tests.blackfennec.extension_system.test_extension_api.test_can_get_view_factory_registry(extension_api,
                                                                    view_factory_regis)
tests.blackfennec.extension_system.test_extension_api.type_loader()
tests.blackfennec.extension_system.test_extension_api.type_registry()
tests.blackfennec.extension_system.test_extension_api.ui_service()
tests.blackfennec.extension_system.test_extension_api.view_factory()
tests.blackfennec.extension_system.test_extension_api.view_factory_registry()

```

tests.blackfennec.extension_system.test_extension_registry module

```

tests.blackfennec.extension_system.test_extension_registry.extension_registry()
tests.blackfennec.extension_system.test_extension_registry.test_can_construct(extension_registry)
tests.blackfennec.extension_system.test_extension_registry.test_can_get_extensions(extension_registry)
tests.blackfennec.extension_system.test_extension_registry.test_can_register_extension(extension_registry)

```

tests.blackfennec.extension_system.test_extension_service module

```

tests.blackfennec.extension_system.test_extension_service.api()
tests.blackfennec.extension_system.test_extension_service.registry()
tests.blackfennec.extension_system.test_extension_service.service(request)
tests.blackfennec.extension_system.test_extension_service.test_can_load_extensions(service,
                                                                    registry,
                                                                    api)
tests.blackfennec.extension_system.test_extension_service.test_can_load_extensions_with_complex_dependencies(service,
                                                                    registry,
                                                                    api)
tests.blackfennec.extension_system.test_extension_service.test_can_load_extensions_with_dependencies(service,
                                                                    registry,
                                                                    api)

```

tests.blackfennec.extension_system.test_extension_service.test_can_load_extensions_with_failing_extensions

tests.blackfennec.extension_system.test_extension_service.test_can_load_extensions_with_missing_dependencies

tests.blackfennec.extension_system.test_presenter_registry module

presenter-registry
def setUp(self):

```
    self.structure_type_name = 'Boolean' self.default_value = False self.alternative_value = True
```

def create_structure(self, value):

```
    return Boolean(value) Tests.
```

This module contains the unit-tests of the presenter-registry.

class tests.blackfennec.extension_system.test_presenter_registry.PresenterRegistryTestSuite(*methodName=*

Bases: TestCase

test_create_presenter_registry()

test_deregister_view()

test_register_view()

tests.blackfennec.extension_system.test_view_factory module

tests.blackfennec.extension_system.test_view_factory.factory(*view*)

tests.blackfennec.extension_system.test_view_factory.test_can_construct(*view_factory*)

tests.blackfennec.extension_system.test_view_factory.test_can_create_view(*view_factory, view*)

tests.blackfennec.extension_system.test_view_factory.view()

tests.blackfennec.extension_system.test_view_factory.view_factory(*factory*)

tests.blackfennec.extension_system.test_view_factory_registry module

tests.blackfennec.extension_system.test_view_factory_registry.factory()

tests.blackfennec.extension_system.test_view_factory_registry.registry()

tests.blackfennec.extension_system.test_view_factory_registry.test_can_create_view_factory_registry(*registry,*

tests.blackfennec.extension_system.test_view_factory_registry.test_can_deregister_view_factory(*registry,*
factory)

tests.blackfennec.extension_system.test_view_factory_registry.test_can_get_factory_if_exists(*registry*,
factory)

tests.blackfennec.extension_system.test_view_factory_registry.test_can_register_view_factory(*registry*,
factory)

tests.blackfennec.extension_system.test_view_factory_registry.test_cannot_deregister_view_factory_if_no

tests.blackfennec.extension_system.test_view_factory_registry.test_cannot_get_factory_if_not_exists(*regi*,
fac-
tory)

Module contents

tests.blackfennec.layers package

Subpackages

tests.blackfennec.layers.encapsulation_base package

Submodules

tests.blackfennec.layers.encapsulation_base.test_base_factory_visitor module

class tests.blackfennec.layers.encapsulation_base.test_base_factory_visitor.BaseFactoryVisitorTestSuite

Bases: TestCase

setUp()

Hook method for setting up the test fixture before exercising it.

test_can_construct()

test_can_visit_boolean()

test_can_visit_list()

test_can_visit_map()

test_can_visit_number()

test_can_visit_reference()

test_can_visit_string()

test_can_visit_structure()

test_double_encapsulation_1(*core_object*)

test_double_encapsulation_2(*core_object*)

```
test_double_encapsulation_3(core_object)
test_double_encapsulation_4(core_object)
test_double_encapsulation_5(core_object)
test_double_encapsulation_6(core_object)
test_encapsulated_can_visit_1(core_object)
test_encapsulated_can_visit_2(core_object)
test_encapsulated_can_visit_3(core_object)
test_encapsulated_can_visit_4(core_object)
test_encapsulated_can_visit_5(core_object)
test_encapsulated_can_visit_6(core_object)
test_visit_caches_class()
```

tests.blackfennec.layers.encapsulation_base.test_encapsulation_base module

```
class tests.blackfennec.layers.encapsulation_base.test_encapsulation_base.EncapsulationBaseTestSuite(me)
```

```
    Bases: TestCase
```

```
    test_parent_getter_1(data)
    test_parent_getter_2(data)
    test_parent_getter_3(data)
    test_parent_getter_4(data)
    test_parent_getter_5(data)
    test_parent_getter_6(data)
    test_parent_setter_1(parent, root, subject, new_parent)
    test_parent_setter_2(parent, root, subject, new_parent)
    test_parent_setter_3(parent, root, subject, new_parent)
    test_parent_setter_4(parent, root, subject, new_parent)
    test_parent_setter_5(parent, root, subject, new_parent)
    test_parent_setter_6(parent, root, subject, new_parent)
    test_root_getter_1(data)
    test_root_getter_2(data)
    test_root_getter_3(data)
    test_root_getter_4(data)
```

```

test_root_getter_5(data)
test_root_getter_6(data)
test_subject_getter_1(data)
test_subject_getter_2(data)
test_subject_getter_3(data)
test_subject_getter_4(data)
test_subject_getter_5(data)
test_subject_getter_6(data)

```

tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base module

```

tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base.layer()
tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base.list_encapsulation_base(layer,
                                                    sub-
                                                    ject)
                                                    →
                                                    Lis-
                                                    tEn-
                                                    cap-
                                                    su-
                                                    la-
                                                    tion-
                                                    Base

tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base.subject()
tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base.test_add_item_item(list_encapsulation_base,
                                                    item)
tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base.test_add_item_item_already_encapsulated(list_encapsulation_base,
                                                                 item)
tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base.test_can_construct(list_encapsulation_base,
                                                                 item)
tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base.test_can_get_repr(list_encapsulation_base,
                                                                 item)
tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base.test_can_get_value_empty(list_encapsulation_base,
                                                                 item)
tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base.test_dispatch_change_notification(list_encapsulation_base,
                                                                 item)

tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base.test_get_value(layer)
tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base.test_remove_encapsulated_item(list_encapsulation_base,
                                                                 item)
tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base.test_remove_item(layer)
tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base.test_replace_encapsulated_item(list_encapsulation_base,
                                                                 item)

```

tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base.test_replace_item(layer)

tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base.test_set_value(list_encapsulation_base)

tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base.test_subject_getter(list_encapsulation_base, subject)

tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base module

tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base.layer()

tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base.map_encapsulation_base(layer, subject) → MapEncapsulationBase

tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base.subject()

tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base.test_add_item_item(map_encapsulation_base, item)

tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base.test_add_item_item_already_encapsulated(map_encapsulation_base, item)

tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base.test_can_construct(map_encapsulation_base)

tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base.test_can_get_repr(map_encapsulation_base)

tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base.test_can_get_value_empty(map_encapsulation_base)

tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base.test_dispatch_change_notification(map_encapsulation_base)

tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base.test_get_value(layer)

tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base.test_remove_item(map_encapsulation_base, subject)

tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base.test_remove_item_not_in_map(map_encapsulation_base)

tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base.test_rename_key(map_encapsulation_base)

tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base.test_replace_item(map_encapsulation_base)

tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base.test_set_value(map_encapsulation_base)

tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base.test_subject_getter(*map_encapsulation_base*, *subject*)

tests.blackfennec.layers.encapsulation_base.test_reference_encapsulation_base module

tests.blackfennec.layers.encapsulation_base.test_reference_encapsulation_base.layer()

tests.blackfennec.layers.encapsulation_base.test_reference_encapsulation_base.test_can_resolve(*layer*)

Module contents

tests.blackfennec.layers.merge package

Submodules

tests.blackfennec.layers.merge.test_integration_merged module

tests.blackfennec.layers.merge.test_integration_merged.map_chain(*layer*, *depth*)

tests.blackfennec.layers.merge.test_integration_merged.test_can_get_child_of_root(*underlay*, *overlay*, *diff*, *expected*)

tests.blackfennec.layers.merge.test_integration_merged.test_can_get_merged_children(*underlay*, *overlay*, *expected*)

tests.blackfennec.layers.merge.test_integration_merged.test_can_get_parent(*underlay*, *overlay*, *expected*)

tests.blackfennec.layers.merge.test_integration_merged.test_can_get_parent_if_no_parent()

tests.blackfennec.layers.merge.test_integration_merged.test_can_get_root(*underlay*, *overlay*, *expected*)

tests.blackfennec.layers.merge.test_integration_merged.test_get_root_returns_merged_structure(*underlay*, *overlay*)

tests.blackfennec.layers.merge.test_integration_merged.test_merge_two_merged_maps_with_null_properties(*underlay*, *overlay*)

tests.blackfennec.layers.merge.test_integration_merged.test_merged_map_null_property(*underlay*, *overlay*)

tests.blackfennec.layers.merge.test_integration_merged.test_merged_parent_is_self()

tests.blackfennec.layers.merge.test_integration_merged.test_only_merge_with_same_type(*underlay*,
over-
lay)

tests.blackfennec.layers.merge.test_integration_merged.test_parent_returns_merged_structure(*underlay*,
over-
lay,
ex-
pected_type)

tests.blackfennec.layers.merge.test_merged_layer module

tests.blackfennec.layers.merge.test_merged_layer.merged(*structure*)

tests.blackfennec.layers.merge.test_merged_layer.parser()

tests.blackfennec.layers.merge.test_merged_layer.structure(*parser*)

tests.blackfennec.layers.merge.test_merged_layer.test_can_access_parent()

tests.blackfennec.layers.merge.test_merged_layer.test_can_access_super(*merged*)

tests.blackfennec.layers.merge.test_merged_layer.test_can_access_super_super(*merged*)

tests.blackfennec.layers.merge.test_merged_layer.test_can_compare_merged_objects()

tests.blackfennec.layers.merge.test_merged_layer.test_can_merge_merged_structure(*merged*)

tests.blackfennec.layers.merge.test_merged_layer.test_can_recover_from_null_merge()

tests.blackfennec.layers.merge.test_merged_layer.test_can_recover_from_null_merge_2()

tests.blackfennec.layers.merge.test_merged_layer.test_cannot_compare_merged_objects_with_different_valu

tests.blackfennec.layers.merge.test_merged_layer.test_cannot_compare_merged_objects_with_standard_objec

tests.blackfennec.layers.merge.test_merged_layer.test_merge_merged_overlay()

tests.blackfennec.layers.merge.test_merged_layer.test_merge_super_first_level(*parser*)

tests.blackfennec.layers.merge.test_merged_layer.test_merge_super_second_level(*parser*)

tests.blackfennec.layers.merge.test_merged_layer.test_merge_with_overlay()

tests.blackfennec.layers.merge.test_merged_layer.test_merges_properties(*parser*)

tests.blackfennec.layers.merge.test_merged_layer.test_navigation_in_doubly_merged_object()

tests.blackfennec.layers.merge.test_merged_layer.test_navigation_in_merged_object()

tests.blackfennec.layers.merge.test_merged_layer.test_navigation_in_phantom_parents()

tests.blackfennec.layers.merge.test_merged_list module

```
tests.blackfennec.layers.merge.test_merged_list.test_can_construct()
tests.blackfennec.layers.merge.test_merged_list.test_can_get_value()
tests.blackfennec.layers.merge.test_merged_list.test_cannot_set_value()
tests.blackfennec.layers.merge.test_merged_list.test_repr()
```

tests.blackfennec.layers.merge.test_merged_map module

```
tests.blackfennec.layers.merge.test_merged_map.test_can_construct()
tests.blackfennec.layers.merge.test_merged_map.test_can_get_value()
tests.blackfennec.layers.merge.test_merged_map.test_cannot_set_value()
tests.blackfennec.layers.merge.test_merged_map.test_repr()
```

tests.blackfennec.layers.merge.test_merged_null module

```
tests.blackfennec.layers.merge.test_merged_null.test_can_accept_visitor()
tests.blackfennec.layers.merge.test_merged_null.test_can_construct()
tests.blackfennec.layers.merge.test_merged_null.test_can_get_subject()
tests.blackfennec.layers.merge.test_merged_null.test_can_get_value()
tests.blackfennec.layers.merge.test_merged_null.test_cannot_construct_with_none_null(underlay,
                                                                                   overlay)
tests.blackfennec.layers.merge.test_merged_null.test_cannot_set_value()
tests.blackfennec.layers.merge.test_merged_null.test_repr()
```

tests.blackfennec.layers.merge.test_merged_phantom module**tests.blackfennec.layers.merge.test_merged_structure module**

```
tests.blackfennec.layers.merge.test_merged_structure.can_get_parent_parameters()
tests.blackfennec.layers.merge.test_merged_structure.can_get_structure_parameters()
tests.blackfennec.layers.merge.test_merged_structure.test_can_accept_correctly(merged)
tests.blackfennec.layers.merge.test_merged_structure.test_can_get_parent(underlay, overlay,
                                                                           expected)
```

```
tests.blackfennec.layers.merge.test_merged_structure.test_can_get_structure(underlay,  
overlay,  
expected)
```

```
tests.blackfennec.layers.merge.test_merged_structure.test_can_get_value(underlay, overlay,  
expected)
```

```
tests.blackfennec.layers.merge.test_merged_structure.test_cannot_set_parent()
```

```
tests.blackfennec.layers.merge.test_merged_structure.test_cannot_set_value()
```

Module contents

tests.blackfennec.layers.observable package

Submodules

tests.blackfennec.layers.observable.test_integration_observable_layer module

```
class tests.blackfennec.layers.observable.test_integration_observable_layer.Action(value,  
names=None,  
*, module=None,  
quality=None,  
name=None,  
type=None,  
start=1,  
boundary=None)
```

Bases: Enum

ADD_ITEM = 4

CHANGE = 3

REDO = 2

REMOVE_ITEM = 5

RESULT = 6

UNDO = 1

```
tests.blackfennec.layers.observable.test_integration_observable_layer.historized(structure,  
observable_layer)
```

```
tests.blackfennec.layers.observable.test_integration_observable_layer.history()
```

```
tests.blackfennec.layers.observable.test_integration_observable_layer.layer(historized,  
request)
```

```
tests.blackfennec.layers.observable.test_integration_observable_layer.observable_layer(history)
```


tests.blackfennec.layers.observable.test_integration_observable_layer.random_string()

tests.blackfennec.layers.observable.test_integration_observable_layer.structure()

tests.blackfennec.layers.observable.test_integration_observable_layer.test_can_create_layer(*historized*)

tests.blackfennec.layers.observable.test_integration_observable_layer.test_can_undo_add_on_list(*history,*
his-
tor-
ized)

tests.blackfennec.layers.observable.test_integration_observable_layer.test_can_undo_modification_on_numl

tests.blackfennec.layers.observable.test_integration_observable_layer.test_can_undo_modification_on_str

tests.blackfennec.layers.observable.test_integration_observable_layer.test_can_undo_redo_on_list(*history,*
layer,
sce-
nario)

tests.blackfennec.layers.observable.test_integration_observable_layer.test_can_undo_redo_on_map(*history,*
layer,
sce-
nario)

tests.blackfennec.layers.observable.test_integration_observable_layer.test_can_undo_redo_on_top_level_m

tests.blackfennec.layers.observable.test_observable module

tests.blackfennec.layers.observable.test_observable.layer()

tests.blackfennec.layers.observable.test_observable.test_can_apply(*layer*)

tests.blackfennec.layers.observable.test_observable.test_can_construct(*layer*)

tests.blackfennec.layers.observable.test_observable.test_can_remember_structure(*layer*)

tests.blackfennec.layers.observable.test_observable_base module

tests.blackfennec.layers.observable.test_observable_base.test_can_construct()

tests.blackfennec.layers.observable.test_observable_base.test_does_append_to_history()

tests.blackfennec.layers.observable.test_observable_entry module

tests.blackfennec.layers.observable.test_observable_entry.test_can_construct()

tests.blackfennec.layers.observable.test_observable_factory_visitor module

tests.blackfennec.layers.observable.test_observable_factory_visitor.history()

tests.blackfennec.layers.observable.test_observable_factory_visitor.test_can_construct(*visitor*)

tests.blackfennec.layers.observable.test_observable_factory_visitor.visitor(*history*)

Module contents

tests.blackfennec.layers.overlay package

Submodules

tests.blackfennec.layers.overlay.test_integration_overlay module

tests.blackfennec.layers.overlay.test_integration_overlay.layer()

tests.blackfennec.layers.overlay.test_integration_overlay.test_can_resolve_double_reference(*layer*)

tests.blackfennec.layers.overlay.test_overlay module

tests.blackfennec.layers.overlay.test_overlay.overlay()

tests.blackfennec.layers.overlay.test_overlay.test_can_apply(*overlay*)

tests.blackfennec.layers.overlay.test_overlay.test_can_construct(*overlay*)

tests.blackfennec.layers.overlay.test_overlay.test_can_remember_structure(*overlay*)

tests.blackfennec.layers.overlay.test_overlay.test_does_not_remember_reference(*overlay*)

tests.blackfennec.layers.overlay.test_overlay_base module

tests.blackfennec.layers.overlay.test_overlay_base.overlay_base()

tests.blackfennec.layers.overlay.test_overlay_base.test_can_accept_and_return_self(*overlay_base*)

tests.blackfennec.layers.overlay.test_overlay_base.test_can_construct(*overlay_base*)

tests.blackfennec.layers.overlay.test_overlay_base.test_can_get_repr(*overlay_base*)

tests.blackfennec.layers.overlay.test_overlay_factory_visitor module

class tests.blackfennec.layers.overlay.test_overlay_factory_visitor.**OverlayFactoryVisitorTestSuite**(*metho*

Bases: TestCase

setUp()

Hook method for setting up the test fixture before exercising it.

test_can_construct()

test_can_visit_structure()

test_generic_overlay_subject()

test_visit_caches_class()

Module contents**Module contents****tests.blackfennec.presentation_system package****Subpackages****tests.blackfennec.presentation_system.main_window package****Submodules****tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model module**

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.**document()**

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.**document_tab**(*document*)

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.**extension_source_registry**

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.**interpretation_service()**

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.**presenter_registry()**

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.**service_locator()**

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.**test_can_attach_tab**(*view_*

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.**test_can_close_file**(*view_*
doc-
u-
ment_

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.**test_can_copy_view_model**

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.**test_can_detach_tab**(*view_*

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.test_can_get_about_window

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.test_can_handle_uri(*view_model*, *service_locator*, *presentation_tab*)

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.test_can_open(*file_path*, *view_model*)

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.test_can_open_file(*view_model*, *file_path*)

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.test_can_redo(*view_model*, *document*, *document_tab*)

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.test_can_save_as_file(*view_model*, *document*, *file_path*, *tmp_dir*)

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.test_can_save_file(*view_model*, *document*, *file_path*, *tmp_dir*)

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.test_can_set_directory(*view_model*, *directory_path*)

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.test_can_undo(*view_model*, *document*, *document_tab*)

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.test_cannot_attach_tab_to_document

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.test_cannot_detach_tab_from_document

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.test_cannot_handle_uri(*view_model*, *service_locator*, *presentation_tab*)

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.test_cannot_redo(*view_model*, *document*, *document_tab*)

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.test_cannot_undo(*view_model*, *document*, *document_tab*)

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.test_save_all(*view_model*, *document*, *document_tab*)

tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model.view_model(*service_locator*)

tests.blackfennec.presentation_system.main_window.test_document_tab module

tests.blackfennec.presentation_system.main_window.test_document_tab.**document**()

tests.blackfennec.presentation_system.main_window.test_document_tab.**document_factory**(*document*)

tests.blackfennec.presentation_system.main_window.test_document_tab.**document_tab**(*presenter_registry*,
document_factory,
navigation_service)

tests.blackfennec.presentation_system.main_window.test_document_tab.**document_tab_parametrized**(*request*,
presenter_registry,
navigation_service)

tests.blackfennec.presentation_system.main_window.test_document_tab.**navigation_service**()

tests.blackfennec.presentation_system.main_window.test_document_tab.**presenter_registry**()

tests.blackfennec.presentation_system.main_window.test_document_tab.**test_can_construct_document_tab**(*document_factory*)

tests.blackfennec.presentation_system.main_window.test_document_tab.**test_can_create_presenter**(*document_factory*,
navigation_service)

tests.blackfennec.presentation_system.main_window.test_document_tab.**test_can_load_document**(*document_tab*,
document_factory)

tests.blackfennec.presentation_system.main_window.test_document_tab.**test_can_save**(*document_tab*,
document)

tests.blackfennec.presentation_system.main_window.test_document_tab.**test_can_save_as**(*tmp_path*,
document_tab_parametrized,
Doc-
u-
ment-
Tab)

Module contents

tests.blackfennec.presentation_system.navigation_service package

Submodules

tests.blackfennec.presentation_system.navigation_service.test_navigation_proxy module

```
class tests.blackfennec.presentation_system.navigation_service.test_navigation_proxy.NavigationProxyTest
    Bases: TestCase
    test_create_navigation_proxy()
    test_navigate()
```

tests.blackfennec.presentation_system.navigation_service.test_navigation_service module

```
class tests.blackfennec.presentation_system.navigation_service.test_navigation_service.NavigationServiceTest
    Bases: TestCase
    test_create_navigation_service()
    test_navigate()
    test_navigate_without_presenter()
```

Module contents

tests.blackfennec.presentation_system.ui_service package

Submodules

tests.blackfennec.presentation_system.ui_service.test_message module

```
tests.blackfennec.presentation_system.ui_service.test_message.message()
tests.blackfennec.presentation_system.ui_service.test_message.test_can_construct(message)
tests.blackfennec.presentation_system.ui_service.test_message.test_can_get_action_name(message)
tests.blackfennec.presentation_system.ui_service.test_message.test_can_get_action_target(message)
tests.blackfennec.presentation_system.ui_service.test_message.test_can_get_text(message)
tests.blackfennec.presentation_system.ui_service.test_message.test_can_get_timeout(message)
```

tests.blackfennec.presentation_system.ui_service.test_ui_service module

tests.blackfennec.presentation_system.ui_service.test_ui_service.**context**(*root*)

tests.blackfennec.presentation_system.ui_service.test_ui_service.**message**()

tests.blackfennec.presentation_system.ui_service.test_ui_service.**message_overlay**(*root*)

tests.blackfennec.presentation_system.ui_service.test_ui_service.**root**()

tests.blackfennec.presentation_system.ui_service.test_ui_service.**test_can_construct**(*ui_service*)

tests.blackfennec.presentation_system.ui_service.test_ui_service.**test_can_deregister_message_overlay**(*ui_service*,
message_overlay)

tests.blackfennec.presentation_system.ui_service.test_ui_service.**test_can_register_message_overlay**(*ui_service*,
message_overlay)

tests.blackfennec.presentation_system.ui_service.test_ui_service.**test_can_show_action_message**(*ui_service*,
message_overlay,
context,
message)

tests.blackfennec.presentation_system.ui_service.test_ui_service.**test_can_show_simple_message**(*ui_service*,
message_overlay,
context,
message)

tests.blackfennec.presentation_system.ui_service.test_ui_service.**ui_service**()

Module contents**Submodules****tests.blackfennec.presentation_system.test_history_service module**

class tests.blackfennec.presentation_system.test_history_service.**Action**(*value*, *names=None*,
module=None,
qualname=None,
type=None, *start=1*,
boundary=None)

Bases: Enum

CHANGE = 3

REDO = 2

RESULT = 4

UNDO = 1

tests.blackfennec.presentation_system.test_history_service.**history**()

tests.blackfennec.presentation_system.test_history_service.**random_string**()

tests.blackfennec.presentation_system.test_history_service.**structure**(*history*)

tests.blackfennec.presentation_system.test_history_service.**test_can_append**(*history, structure*)

tests.blackfennec.presentation_system.test_history_service.**test_can_construct**(*history*)

tests.blackfennec.presentation_system.test_history_service.**test_can_redo**(*history*)

tests.blackfennec.presentation_system.test_history_service.**test_can_tell_if_can_redo**(*history, structure*)

tests.blackfennec.presentation_system.test_history_service.**test_can_undo**(*history*)

tests.blackfennec.presentation_system.test_history_service.**test_can_undo_redo**(*structure, history, scenario*)

Module contents

tests.blackfennec.structure package

Subpackages

tests.blackfennec.structure.reference_navigation package

Submodules

tests.blackfennec.structure.reference_navigation.test_child_navigator module

tests.blackfennec.structure.reference_navigation.test_child_navigator.**list**()

tests.blackfennec.structure.reference_navigation.test_child_navigator.**map**()

tests.blackfennec.structure.reference_navigation.test_child_navigator.**test_can_construct**()

tests.blackfennec.structure.reference_navigation.test_child_navigator.**test_get_representation**()

tests.blackfennec.structure.reference_navigation.test_child_navigator.**test_hash_equal_tokens**()

tests.blackfennec.structure.reference_navigation.test_child_navigator.**test_hash_not_equal_tokens**()

tests.blackfennec.structure.reference_navigation.test_child_navigator.**test_navigate_invalid_type**()


```

tests.blackfennec.structure.reference_navigation.test_child_navigator.test_navigate_list(list,
                                                                                          child_index)
tests.blackfennec.structure.reference_navigation.test_child_navigator.test_navigate_list_invalid_index(
tests.blackfennec.structure.reference_navigation.test_child_navigator.test_navigate_map(map,
                                                                                          child_key)
tests.blackfennec.structure.reference_navigation.test_child_navigator.test_navigate_map_invalid_key(map
tests.blackfennec.structure.reference_navigation.test_child_navigator.test_navigators_are_equal()
tests.blackfennec.structure.reference_navigation.test_child_navigator.test_navigators_are_not_equal()

```

tests.blackfennec.structure.reference_navigation.test_index_of_navigator module

```

tests.blackfennec.structure.reference_navigation.test_index_of_navigator.index_of_navigator()
tests.blackfennec.structure.reference_navigation.test_index_of_navigator.list()
tests.blackfennec.structure.reference_navigation.test_index_of_navigator.map()
tests.blackfennec.structure.reference_navigation.test_index_of_navigator.test_can_construct(index_of_navig
tests.blackfennec.structure.reference_navigation.test_index_of_navigator.test_get_representation(index_of
tests.blackfennec.structure.reference_navigation.test_index_of_navigator.test_navigate_from_child_witho
tests.blackfennec.structure.reference_navigation.test_index_of_navigator.test_navigate_list(list,
                                                                                          child_index)
tests.blackfennec.structure.reference_navigation.test_index_of_navigator.test_navigate_map(map,
                                                                                          child_key)

```

tests.blackfennec.structure.reference_navigation.test_navigator module

```
class tests.blackfennec.structure.reference_navigation.test_navigator.NavigatorSubclass
```

Bases: *Navigator*

navigate(*structure*)

navigates current structure and returns destination

Returns

Structure navigated to

Return type

Structure

Raises

NotImplementedError – if subclass did not implement this method

```

tests.blackfennec.structure.reference_navigation.test_navigator.navigator()
tests.blackfennec.structure.reference_navigation.test_navigator.test_navigator_equality(navigator)
tests.blackfennec.structure.reference_navigation.test_navigator.test_navigator_hash(navigator)
tests.blackfennec.structure.reference_navigation.test_navigator.test_navigator_inequality(navigator)

```

tests.blackfennec.structure.reference_navigation.test_parent_navigator module

```
tests.blackfennec.structure.reference_navigation.test_parent_navigator.child(parent)
tests.blackfennec.structure.reference_navigation.test_parent_navigator.parent()
tests.blackfennec.structure.reference_navigation.test_parent_navigator.parent_navigator()
tests.blackfennec.structure.reference_navigation.test_parent_navigator.test_can_construct(parent_navigator)
tests.blackfennec.structure.reference_navigation.test_parent_navigator.test_get_representation(parent_navigator)
tests.blackfennec.structure.reference_navigation.test_parent_navigator.test_navigate(parent_navigator,
                                             parent,
                                             child)
```

tests.blackfennec.structure.reference_navigation.test_root_navigator module

```
tests.blackfennec.structure.reference_navigation.test_root_navigator.root()
tests.blackfennec.structure.reference_navigation.test_root_navigator.root_child(root)
tests.blackfennec.structure.reference_navigation.test_root_navigator.root_navigator()
tests.blackfennec.structure.reference_navigation.test_root_navigator.test_can_construct(root_navigator)
tests.blackfennec.structure.reference_navigation.test_root_navigator.test_get_representation(root_navigator)
tests.blackfennec.structure.reference_navigation.test_root_navigator.test_navigate(root_navigator,
                                             root,
                                             root_child)
```

tests.blackfennec.structure.reference_navigation.test_sibling_offset_navigator module

```
tests.blackfennec.structure.reference_navigation.test_sibling_offset_navigator.structure()
tests.blackfennec.structure.reference_navigation.test_sibling_offset_navigator.test_can_construct()
tests.blackfennec.structure.reference_navigation.test_sibling_offset_navigator.test_failing_navigate_in()
tests.blackfennec.structure.reference_navigation.test_sibling_offset_navigator.test_get_representation()
tests.blackfennec.structure.reference_navigation.test_sibling_offset_navigator.test_hash_of_navigators()
tests.blackfennec.structure.reference_navigation.test_sibling_offset_navigator.test_hash_of_navigators()
tests.blackfennec.structure.reference_navigation.test_sibling_offset_navigator.test_navigate(structure,
                                                 offset,
                                                 child_index)
tests.blackfennec.structure.reference_navigation.test_sibling_offset_navigator.test_navigators_with_diff()
tests.blackfennec.structure.reference_navigation.test_sibling_offset_navigator.test_navigators_with_same()
```

tests.blackfennec.structure.reference_navigation.test_uri_navigator module

```

tests.blackfennec.structure.reference_navigation.test_uri_navigator.document()
tests.blackfennec.structure.reference_navigation.test_uri_navigator.document_factory(document)
tests.blackfennec.structure.reference_navigation.test_uri_navigator.root(document)
tests.blackfennec.structure.reference_navigation.test_uri_navigator.structure(root)
tests.blackfennec.structure.reference_navigation.test_uri_navigator.test_can_construct(uri_navigator)
tests.blackfennec.structure.reference_navigation.test_uri_navigator.test_get_representation(uri_navigator)
tests.blackfennec.structure.reference_navigation.test_uri_navigator.test_hash_of_navigators_with_differ
tests.blackfennec.structure.reference_navigation.test_uri_navigator.test_hash_of_navigators_with_same_o
tests.blackfennec.structure.reference_navigation.test_uri_navigator.test_navigate(uri_navigator,
                                                                              structure,
                                                                              docu-
                                                                              ment_factory,
                                                                              docu-
                                                                              ment)
tests.blackfennec.structure.reference_navigation.test_uri_navigator.test_navigators_with_different_offs
tests.blackfennec.structure.reference_navigation.test_uri_navigator.test_navigators_with_same_offset_ar
tests.blackfennec.structure.reference_navigation.test_uri_navigator.uri_navigator(document_factory)

```

Module contents**Submodules****tests.blackfennec.structure.test_boolean module**

```
class tests.blackfennec.structure.test_boolean.BooleanTestSuite(methodName='runTest')
```

```
    Bases: StructureTestMixin, TestCase
```

```
    create_instance(value)
```

```
    setUp()
```

```
        Hook method for setting up the test fixture before exercising it.
```

```
    test_accept()
```

```
    test_can_change_parent()
```

```
    test_can_construct()
```

```
    test_can_default_construct()
```

```
    test_can_get_value()
```

```
    test_can_set_value()
```

```
    test_representation()
```

tests.blackfennec.structure.test_list module

tests.blackfennec.structure.test_list.test_accept()
tests.blackfennec.structure.test_list.test_add_item_does_set_parent()
tests.blackfennec.structure.test_list.test_add_item_logs_on_parent_not_none()
tests.blackfennec.structure.test_list.test_add_item_throws_on_parent_not_none()
tests.blackfennec.structure.test_list.test_can_add_all_core_types(*create_structure*)
tests.blackfennec.structure.test_list.test_can_add_item_item()
tests.blackfennec.structure.test_list.test_can_construct()
tests.blackfennec.structure.test_list.test_can_construct_from_list()
tests.blackfennec.structure.test_list.test_can_get_value()
tests.blackfennec.structure.test_list.test_can_remove_item_item()
tests.blackfennec.structure.test_list.test_can_set_value()
tests.blackfennec.structure.test_list.test_notifies_on_item_removal()
tests.blackfennec.structure.test_list.test_notifies_on_value_change()
tests.blackfennec.structure.test_list.test_remove_item_does_unset_parent()
tests.blackfennec.structure.test_list.test_remove_item_throws_on_delete_of_not_existing_item()

tests.blackfennec.structure.test_map module

tests.blackfennec.structure.test_map.test_accept()
tests.blackfennec.structure.test_map.test_add_item_does_set_parent()
tests.blackfennec.structure.test_map.test_add_item_logs_on_key_occupied(*caplog*)
tests.blackfennec.structure.test_map.test_add_item_raises_assertion_error()
tests.blackfennec.structure.test_map.test_add_item_throws_on_key_occupied()
tests.blackfennec.structure.test_map.test_add_item_throws_on_parent_not_none()
tests.blackfennec.structure.test_map.test_can_add_all_core_types(*create_structure*)
tests.blackfennec.structure.test_map.test_can_add_item()
tests.blackfennec.structure.test_map.test_can_construct()
tests.blackfennec.structure.test_map.test_can_construct_from_dict()
tests.blackfennec.structure.test_map.test_can_get_value()
tests.blackfennec.structure.test_map.test_can_remove_item()

```

tests.blackfennec.structure.test_map.test_can_set_value()
tests.blackfennec.structure.test_map.test_can_set_value_when_map_has_content()
tests.blackfennec.structure.test_map.test_logs_on_remove_item_not_existing(caplog)
tests.blackfennec.structure.test_map.test_notifies_on_item_add()
tests.blackfennec.structure.test_map.test_notifies_on_item_removal()
tests.blackfennec.structure.test_map.test_notifies_on_value_change()
tests.blackfennec.structure.test_map.test_remove_item_does_unset_parent()
tests.blackfennec.structure.test_map.test_rename_key()
tests.blackfennec.structure.test_map.test_rename_key_notifies_only_once()
tests.blackfennec.structure.test_map.test_replace_item()
tests.blackfennec.structure.test_map.test_replace_item_notifies_only_once()
tests.blackfennec.structure.test_map.test_throws_on_remove_item_not_existing()

```

tests.blackfennec.structure.test_null module

```

class tests.blackfennec.structure.test_null.NullTestSuite(methodName='runTest')
    Bases: StructureTestMixin, TestCase
        create_instance(value=None)

        setUp()
            Hook method for setting up the test fixture before exercising it.

        test_can_construct()

        test_can_default_construct()

```

tests.blackfennec.structure.test_number module

```

class tests.blackfennec.structure.test_number.NumberTestSuite(methodName='runTest')
    Bases: StructureTestMixin, TestCase
        create_instance(value)

        setUp()
            Hook method for setting up the test fixture before exercising it.

        test_can_construct()

        test_can_default_construct()

        test_can_remove_equal_numbers_from_map()

        test_cannot_check_if_number_in_list()

        test_cannot_find_number_in_list()

```

tests.blackfennec.structure.test_reference module

```
tests.blackfennec.structure.test_reference.navigator(structure)
tests.blackfennec.structure.test_reference.reference(navigator)
tests.blackfennec.structure.test_reference.structure()
tests.blackfennec.structure.test_reference.test_can_construct(reference)
tests.blackfennec.structure.test_reference.test_can_set_value(reference, navigator)
tests.blackfennec.structure.test_reference.test_notifies_on_value_change(reference, navigator)
tests.blackfennec.structure.test_reference.test_resolve(reference, navigator, structure)
```

tests.blackfennec.structure.test_string module

```
class tests.blackfennec.structure.test_string.StringTestSuite(methodName='runTest')
    Bases: StructureTestMixin, TestCase
    create_instance(value)
    setUp()
        Hook method for setting up the test fixture before exercising it.
    test_can_construct()
    test_can_default_construct()
```

tests.blackfennec.structure.test_structure module

```
class tests.blackfennec.structure.test_structure.StructureTestMixin
    Bases: object
    abstract create_instance(value)
    test_can_accept()
    test_can_get_value()
    test_can_set_parent()
    test_can_set_value()
    test_notifies_on_value_change()
```

Module contents

`tests.blackfennec.type_system` package

Subpackages

`tests.blackfennec.type_system.interpretation` package

Submodules

`tests.blackfennec.type_system.interpretation.test_coverage` module

```

class tests.blackfennec.type_system.interpretation.test_coverage.CoverageTestSuite(methodName='runTest')
    Bases: TestCase
        test_can_construct_not_covered()
        test_equal_coverages_equality()
        test_is_covered_false()
        test_is_covered_true()
        test_lower_than_equal()
        test_lower_than_lower_and_greater()
        test_not_equal_coverages_equality()

```

`tests.blackfennec.type_system.interpretation.test_integration_interpretation_service` module

```

tests.blackfennec.type_system.interpretation.test_integration_interpretation_service.test_auction(types,
structure,
type)

```

`tests.blackfennec.type_system.interpretation.test_integration_interpretation_service.types()`

`tests.blackfennec.type_system.interpretation.test_interpretation` module

InterpretationService Tests.

This module contains the unit-tests of the Interpretation class.

```

class tests.blackfennec.type_system.interpretation.test_interpretation.InterpretationTestSuite(methodName='runTest')
    Bases: TestCase
        test_create_interpretation()
        test_navigation_destination_is_argument()
        test_navigation_request_is_dispatched()

```

`test_navigation_sender_is_interpretation()`

`test_set_navigation_service()`

`test_structure_getter()`

tests.blackfennec.type_system.interpretation.test_interpretation_service module

`tests.blackfennec.type_system.interpretation.test_interpretation_service.interpreter(type_registry)`

`tests.blackfennec.type_system.interpretation.test_interpretation_service.test_all_satisfying_offers_are`

`tests.blackfennec.type_system.interpretation.test_interpretation_service.test_auction_with_no_fitting_o`

`tests.blackfennec.type_system.interpretation.test_interpretation_service.test_can_create_interpretation`

`tests.blackfennec.type_system.interpretation.test_interpretation_service.test_create_interpreter(interpre`

`tests.blackfennec.type_system.interpretation.test_interpretation_service.test_only_satisfying_offers_ar`

`tests.blackfennec.type_system.interpretation.test_interpretation_service.type_registry()`

tests.blackfennec.type_system.interpretation.test_offer module

`tests.blackfennec.type_system.interpretation.test_offer.offer(subject, type)`

`tests.blackfennec.type_system.interpretation.test_offer.subject()`

`tests.blackfennec.type_system.interpretation.test_offer.test_can_create_offer(offer)`

`tests.blackfennec.type_system.interpretation.test_offer.test_coverage_getter_simple(subject)`

`tests.blackfennec.type_system.interpretation.test_offer.test_equal_offers_equality(subject)`

`tests.blackfennec.type_system.interpretation.test_offer.test_lower_than_equal(offer, subject,
type)`

`tests.blackfennec.type_system.interpretation.test_offer.test_lower_than_with_different_subject(offer,
type)`

`tests.blackfennec.type_system.interpretation.test_offer.test_representation(offer)`

`tests.blackfennec.type_system.interpretation.test_offer.test_respects_inheritance_hierarchy(offer,
sub-
ject,
type)`

`tests.blackfennec.type_system.interpretation.test_offer.test_specificity_getter(offer)`

`tests.blackfennec.type_system.interpretation.test_offer.test_subject_getter(offer, subject)`

`tests.blackfennec.type_system.interpretation.test_offer.test_type_getter(offer, type)`

`tests.blackfennec.type_system.interpretation.test_offer.type()`

tests.blackfennec.type_system.interpretation.test_specification module

```
class tests.blackfennec.type_system.interpretation.test_specification.SpecificationTestSuite(methodName)
    Bases: TestCase
    test_can_default_create()
    test_can_request_preview()
    test_representation()
```

Module contents**Submodules****tests.blackfennec.type_system.test_boolean_type module**

```
class tests.blackfennec.type_system.test_boolean_type.TestBooleanType
    Bases: object
    test_can_construct(type)
    test_default(type: BooleanType)
    test_minimum_default(type: BooleanType)
    test_omitted_default()
    test_repr(type: BooleanType)
    test_reuses_structure_for_expected(type)
    test_set_expected_to_false(type: BooleanType)
    test_set_expected_to_true(type: BooleanType)
    test_visitor(type: BooleanType)
    test_visitor_expected_false(type: BooleanType)
    test_visitor_expected_true(type: BooleanType)
    type() → BooleanType
```

tests.blackfennec.type_system.test_integration_type_loader module

```
tests.blackfennec.type_system.test_integration_type_loader.document_factory(type_registry) →
    DocumentFactory
tests.blackfennec.type_system.test_integration_type_loader.test_for_subclass_property(type)
tests.blackfennec.type_system.test_integration_type_loader.test_for_superclass_property(type)
```

```
tests.blackfennec.type_system.test_integration_type_loader.test_inheritance_in_inherited(type)
tests.blackfennec.type_system.test_integration_type_loader.test_inherits_maximum(type)
tests.blackfennec.type_system.test_integration_type_loader.test_merge_null(merger, structure)
tests.blackfennec.type_system.test_integration_type_loader.test_merges_recursively(type)
tests.blackfennec.type_system.test_integration_type_loader.test_overrides_default(type)
tests.blackfennec.type_system.test_integration_type_loader.test_overrides_minimum(type)
tests.blackfennec.type_system.test_integration_type_loader.test_overrides_nested_constraintes(type)
tests.blackfennec.type_system.test_integration_type_loader.test_type_covers_good_instance(type)
tests.blackfennec.type_system.test_integration_type_loader.type(tmp_path, document_factory,
                                                             type_registry)
tests.blackfennec.type_system.test_integration_type_loader.type_registry()
```

tests.blackfennec.type_system.test_list_type module

```
class tests.blackfennec.type_system.test_list_type.ListTypeTestSuite(methodName='runTest')
```

```
    Bases: TestCase
```

```
    setUp()
```

```
        Hook method for setting up the test fixture before exercising it.
```

```
    test_calculate_coverage_wrong_type()
```

```
    test_can_add_element()
```

```
    test_can_add_optional_element()
```

```
    test_can_add_required_element()
```

```
    test_can_construct()
```

```
    test_can_create_instance()
```

```
    test_can_get_repr()
```

```
    test_can_make_index_optional()
```

```
    test_can_make_index_required()
```

```
    test_can_set_child_optional()
```

```
    test_can_tell_that_child_is_optional()
```

```
    test_can_tell_that_child_is_required()
```

```
    test_cannot_set_none_child_optionality()
```

```
    test_coverage_getter_list_full_coverage()
```

```
    test_coverage_getter_list_half_coverage()
```

tests.blackfennec.type_system.test_map_type module

class tests.blackfennec.type_system.test_map_type.**MapTypeTestSuite**(*methodName='runTest'*)

Bases: TestCase

setUp()

Hook method for setting up the test fixture before exercising it.

test_calculate_coverage_map_full_coverage()

test_calculate_coverage_map_half_coverage()

test_calculate_coverage_map_incompatible()

test_calculate_coverage_map_third_coverage()

test_calculate_coverage_wrong_type()

test_can_construct()

test_can_create_instance()

test_can_get_repr()

test_can_recognize_self()

test_can_set_child_optional()

test_can_set_required_to_false()

test_can_set_required_to_true()

test_can_tell_that_child_is_optional()

test_can_tell_that_child_is_required()

test_cannot_set_none_child_optionality()

test_cannot_set_required_for_none_property()

test_required_properties()

tests.blackfennec.type_system.test_null_type module

tests.blackfennec.type_system.test_null_type.**test_can_construct**(*type*)

tests.blackfennec.type_system.test_null_type.**test_default**(*type: NullType*)

tests.blackfennec.type_system.test_null_type.**test_repr**(*type: NullType*)

tests.blackfennec.type_system.test_null_type.**test_visitor**(*type: NullType*)

tests.blackfennec.type_system.test_null_type.**type**() → *NullType*

tests.blackfennec.type_system.test_number_type module

```
class tests.blackfennec.type_system.test_number_type.TestNumberType
    Bases: object
    test_can_be_created(type)
    test_can_get_repr(type)
    test_can_reset_limits(type: NumberType)
    test_default(type: NumberType)
    test_maximum_default(type: NumberType)
    test_minimum_default(type: NumberType)
    test_omitted_default()
    test_omitted_maximum()
    test_omitted_minimum()
    test_set_maximum(type: NumberType)
    test_set_minimum(type: NumberType)
    test_visitor(type: NumberType)
    test_visitor_allows_exactly(type: NumberType)
    test_visitor_checks_maximum(type: NumberType)
    test_visitor_checks_minimum(type: NumberType)
    type() → NumberType
```

tests.blackfennec.type_system.test_string_type module

```
class tests.blackfennec.type_system.test_string_type.StringTypeTestSuite(methodName='runTest')
    Bases: TestCase
    setUp()
        Hook method for setting up the test fixture before exercising it.
    test_calculate_coverage_wrong_type()
    test_can_calculate_coverage()
    test_can_calculate_coverage_normal_string()
    test_can_calculate_coverage_pattern_match()
    test_can_calculate_coverage_pattern_mismatch()
    test_can_construct()
    test_can_cover_string()
```

```
test_can_create_instance()
```

```
test_can_get_repr()
```

tests.blackfennec.type_system.test_type_base module

```
tests.blackfennec.type_system.test_type_base.test_can_calculate_coverage_of_structure(type,  
                                                                                   structure)
```

```
tests.blackfennec.type_system.test_type_base.test_can_construct(type)
```

```
tests.blackfennec.type_system.test_type_base.test_has_create_instance_interface(type)
```

```
tests.blackfennec.type_system.test_type_base.test_optional_getter(type)
```

```
tests.blackfennec.type_system.test_type_base.test_optional_setter(type)
```

```
tests.blackfennec.type_system.test_type_base.type(request)
```

tests.blackfennec.type_system.test_type_parser module

```
tests.blackfennec.type_system.test_type_parser.test_create_type()
```

```
tests.blackfennec.type_system.test_type_parser.test_type_from_json()
```

tests.blackfennec.type_system.test_type_registry module

type-registry Tests.

This module contains the unit-tests of the type-registry.

```
class tests.blackfennec.type_system.test_type_registry.TypeRegistryTestSuite(methodName='runTest')
```

```
    Bases: TestCase
```

```
        test_create_type_registry()
```

```
        test_deregister_view()
```

```
        test_register_view()
```

```
        test_returns_copy_of_internal_list()
```

Module contents

tests.blackfennec.util package

Submodules

tests.blackfennec.util.test_change_notification module

```
tests.blackfennec.util.test_change_notification.change_notification()
tests.blackfennec.util.test_change_notification.test_can_construct(change_notification)
tests.blackfennec.util.test_change_notification.test_get_new_value(change_notification)
tests.blackfennec.util.test_change_notification.test_get_old_value(change_notification)
```

tests.blackfennec.util.test_change_notification_dispatch_mixin module

```
tests.blackfennec.util.test_change_notification_dispatch_mixin.change_notification_dispatch_mixin()
tests.blackfennec.util.test_change_notification_dispatch_mixin.test_can_construct(change_notification_dispatch_mixin)
tests.blackfennec.util.test_change_notification_dispatch_mixin.test_dispatch_change_notification(change_notification_dispatch_mixin)
tests.blackfennec.util.test_change_notification_dispatch_mixin.test_skip_unnecessary_dispatch_of_change_notification_dispatch_mixin()
```

tests.blackfennec.util.test_comparable module

```
class tests.blackfennec.util.test_comparable.ComparableTestSuite(methodName='runTest')
    Bases: TestCase
    test_greater_equal_equal_elements()
    test_greater_equal_greater_element()
    test_greater_equal_lower_element()
    test_greater_than_equal_elements()
    test_greater_than_greater_element()
    test_greater_than_lower_element()
    test_lower_equal_equal_elements()
    test_lower_equal_greater_element()
    test_lower_equal_lower_element()
    test_not_equal_equal_elements()
    test_not_equal_unequal_elements()
```

tests.blackfennec.util.test_deep_copy module

```
class tests.blackfennec.util.test_deep_copy.TestDeepCopyVisitorTestSuite(methodName='runTest')
    Bases: TestCase
    setUp()
        Hook method for setting up the test fixture before exercising it.
    test_visit_boolean()
```

```
test_visit_list()
test_visit_map()
test_visit_number()
test_visit_reference()
test_visit_string()
test_visit_structure()
```

tests.blackfennec.util.test_integration_service_locator module

```
tests.blackfennec.util.test_integration_service_locator.test_can_construct_service_locator()
```

tests.blackfennec.util.test_meta_info module

```
tests.blackfennec.util.test_meta_info.test_get_app_id()
tests.blackfennec.util.test_meta_info.test_get_authors()
tests.blackfennec.util.test_meta_info.test_get_copy_right()
tests.blackfennec.util.test_meta_info.test_get_description()
tests.blackfennec.util.test_meta_info.test_get_home_page()
tests.blackfennec.util.test_meta_info.test_get_icon_path()
tests.blackfennec.util.test_meta_info.test_get_issue_page()
tests.blackfennec.util.test_meta_info.test_get_license()
tests.blackfennec.util.test_meta_info.test_get_name()
tests.blackfennec.util.test_meta_info.test_get_release_notes()
tests.blackfennec.util.test_meta_info.test_get_summary()
tests.blackfennec.util.test_meta_info.test_get_version()
```

tests.blackfennec.util.test_observable module

```
class tests.blackfennec.util.test_observable.ObservableTestSuite(methodName='runTest')
```

```
    Bases: TestCase
```

```
    test_bind_kwargs()
```

Module contents

Module contents

tests.test_utils package

Submodules

tests.test_utils.connection module

`tests.test_utils.connection.has_internet_connection()`

Checks if the system has an internet connection

Returns

True if system has internet connection, False otherwise

Return type

bool

tests.test_utils.deep_compare module

class `tests.test_utils.deep_compare.BooleanComparator` (*boolean*)

Bases: *ComparatorTemplate*

visit_boolean(*other*)

class `tests.test_utils.deep_compare.ComparatorFactory`

Bases: object

Create a comparator using the visitor pattern

visit_boolean(*subject*: Boolean)

visit_list(*subject*: List)

visit_map(*subject*: Map)

visit_null(*unused_subject*: Null)

visit_number(*subject*: Number)

visit_reference(*subject*: Reference)

visit_string(*subject*: String)

visit_structure(*subject*: Structure)

class `tests.test_utils.deep_compare.ComparatorTemplate`

Bases: object

Base Comparator is a Template Pattern and always returns False.

visit_boolean(*unused_other*: Boolean)

visit_list(*unused_other*: List)


```
visit_map(UNUSED_OTHER: Map)
visit_null(UNUSED_OTHER: Null)
visit_number(UNUSED_OTHER: Number)
visit_reference(UNUSED_OTHER: Reference)
visit_string(UNUSED_OTHER: String)
visit_structure(UNUSED_OTHER: Structure)

class tests.test_utils.deep_compare.DeepCompare
  Bases: object
  static compare(structure_a, structure_b)

class tests.test_utils.deep_compare.ListComparator(subject)
  Bases: ComparatorTemplate
  Compare structure to list via visitor pattern.
  visit_list(OTHER)

class tests.test_utils.deep_compare.MapComparator(subject)
  Bases: ComparatorTemplate
  Compare structure to map via visitor pattern.
  visit_map(OTHER)

class tests.test_utils.deep_compare.NullComparator
  Bases: ComparatorTemplate
  visit_null(OTHER)

class tests.test_utils.deep_compare.NumberComparator(number)
  Bases: ComparatorTemplate
  visit_number(OTHER)

class tests.test_utils.deep_compare.ReferenceComparator(subject)
  Bases: ComparatorTemplate
  visit_reference(OTHER)

class tests.test_utils.deep_compare.StringComparator(string)
  Bases: ComparatorTemplate
  visit_string(OTHER)

class tests.test_utils.deep_compare.StructureComparator(UNUSED_SUBJECT)
  Bases: ComparatorTemplate
  visit_structure(UNUSED_OTHER)
```

tests.test_utils.observer module

class tests.test_utils.observer.Observer

Bases: object

property calls

endpoint(*args, **kwargs)

property last_call

tests.test_utils.parameterize module

tests.test_utils.what_the_farmer_does_not_eat_visitor_factory module

class tests.test_utils.what_the_farmer_does_not_eat_visitor_factory.
WhatTheFarmerDoesNotEatVisitorFactory

Bases: object

visit_boolean(unused_subject: Boolean)

visit_list(unused_subject: List)

visit_map(unused_unused_subject: Map)

visit_null(unused_unused_subject: Null)

visit_number(unused_subject: Number)

visit_reference(unused_subject: Reference)

visit_string(unused_subject: String)

visit_structure(unused_subject: Structure)

Module contents

Submodules

tests.context module

Module contents

Py-ModIndex

3.3 Project

In this part of the documentation the project itself is described. This includes the project management, the requirements engineering part and the project standards.

Most of Black Fennec was developed in the scope of two consecutive projects by students of the University of Applied Sciences in Rapperswil (HSR). These two projects including their planning can be found in the *project management*

section. Additionally, more general information such as *design decisions* or the *risk analysis* are also contained in this section.

The requirements engineering contains the *functional* and *non-functional requirements* of the project. The functional requirements are described with an *epic overview* and more specific *user stories* while the *non-functional requirements* follow the ISO 25010 standard.

In the *standards section* the project standards are described. These standards are used to ensure a consistent and high quality code base.

Following is an overview of the contained information in this section:

3.3.1 Project Management

This section contains the two projects which make up most of the initial contributions to Black Fennec. Additionally, you will find an overview of key *design decisions* and a detailed *risk analysis* which have inter-project significance.

The first project is called the *engineering project* and is a student project of the University of Applied Sciences in Rapperswil (HSR). In it four developers cumulatively invested approximately 480 hours of work. It laid the groundwork for the project and was the first step towards a fully functional application.

The second project called *study project* is also a student project of the Eastern Switzerland University of Applied Sciences (OST). It is a continuation of the engineering project and has two developers also cumulatively investing approximately 480 hours of work. It's main goal is the release of Version 1.0 of Black Fennec in a stable and mature manner.

For an overview of the contained subsections please refer to the following table of contents:

Engineering Project

The *engineering project* is a semester long project that is carried out by a group of students. In the context of this project an accumulated 480 hours were invested into the development of Black Fennec.

Contrary to the usual way of selecting a predefined topic that is given by a lecturer, the students came up with the idea for Black Fennec. This was done via a *project proposal* and guided by Thomas Kälin, lecturer at the University of Applied Sciences Rapperswil who supervised the project.

The *project plan* was created at the beginning of the project and provides a rough overview of our project, the project organization, and the management processes. The plan was updated regularly during the project. Additionally to this plan another artifact created was during/after the *elaboration phase* of the project and highlights the most important decisions made during this phase and contains a detailed plan for the succeeding *construction phase*.

The *refinement document* contains suggestions for improvement, that were received, and how they were addressed. Additionally to that this section also contains the projects *time tracking* and a *reflection statement* that was written at the end of the project.

An overview of all the mentioned subsections can be found in the following table of contents:

Proposal

Project Name	Date
Black Fennec	17.12.2020

Team

- Leonie Däullary <leonie.daeullary@ost.ch>
- Lara Gubler <lara.gubler@ost.ch>
- Simon Kindhauser <simon.kindhauser@ost.ch>
- Caspar Martens <caspar.martens@ost.ch>

Consulting and Review Times

<i>Time Slots</i>	Monday	Tuesday	Wednesday	Thursday	Friday
08:00-09:00					XO
09:00-10:00					XO
10:00-11:00					(XO)
11:00-12:00					(XO)
12:00-13:00					(XO)
13:00-14:00					XO
14:00-15:00					XO
15:00-16:00					XO
16:00-17:00					XO
17:00-18:00					(XO)
18:00-19:00					

X = Team is available

(X) = Team is available if need be

O = Meeting online is possible

P = Meeting physically at Campus OST-RJ is possible

= Team is unavailable

Motivation

We personally require an application that is able to manage unstructured data in a dynamically structured way. Additionally it ought to be possible to use versioning and collaborative tools such as git. This is why we want to develop such an application as our Engineering Project. We will use Black Fennec to help us organize and share unstructured data.

Typical Use Case

A headhunter tasked with employing a software engineer for a company found a Github repository of a promising candidate. He will contact the candidate but first wants to know more about her in order to better influence her into taking the job.

For this Marco, the headhunter, creates a new Black Fennec Project. There he can add information already gathered from the Github profile into a pre-existing template. A extension might allow automatic crawling of additional information based on what was previously entered, automating a repetitive task. Some of the new information can now be used to further research the target. All additional data can be added to the project, even if no specific template exists. Once he is done the project can be pushed to a remote server where others can continue working.

Project Idea

Black Fennec is a versionable, extendable, dynamically structured information management tool that is visually appealing and pleasant to use.

Extensibility is generally very important to the project, because these modules are an important part of the work. The base of the project offers only very basic functionalities which we want to extend in the context of this work. This allows us to work independently and scale the effort dynamically.

Project idea discussed with: Thomas Kälin

Implementation

The application is based on YAML/JSON files which are interpreted and visualized in a Python client. The GUI technology is not decided yet, but the plan is to go in the direction of declarative frameworks (XAML/HTML). The extensions will have their own dependencies which need to be installed automatically. The client is optimized for desktop systems.

Technology

As mentioned, we will develop our project using Python. We strive for Test Driven Development and this will be realized with the help of the PyTest module. Our GUI will be implemented using GTK3, which supports Linux, MacOS and Windows. We will use MVVM to separate UI from logic.

Core

The core is the basis of our project and should provide a framework that allows the various extensions to integrate into the system. In addition, it offers certain core features that are generally required. These are in particular the types String, Number, References, List and Dictionary. These types form the basis of the dynamic object model, which is very well suited for our product. The core GUI, which also contains the visualization of the listed types, is also implemented here. Support for the search feature must also be implemented in the core.

Base

Base is the only directly built-in extension. It provides functionality that our users are frequently relying on. For example the types: File, Image, Date, Location. You can also think of Base as a Minimal Viable Product. The goal is to keep it as lean as possible and to outsource any additional functionality to other extensions, so that our application remains as flexible as possible.

Additional Extensions

This list is a incomplete collection of extensions that could be implemented in the future. They are mainly listed here for you to get a feel what might be coming, if we find the time to implement more than just of Black Fennec.

Extension Name	Description
GitHub Crawler	Automated crawling of user data of a GitHub account by specifying the profile URI.
Facebook Crawler	Automated crawling of user data of a Facebook account by specifying the profile URI.
LinkedIn Crawler	Automated crawling of user data of a LinkedIn account by specifying the profile URI.
Facial Recognition	Extraction of faces from images. Possibly referencing to similar faces within the project.
Google Dorks	Generate Google Dorks from selected Types.

Plan

Introduction

Objective

The purpose of this document is to describe the project plan and provide an overview of the Black Fennec project. This document contains a rough overview of our project, the project organization, and management processes. It also includes risk management and some brief information about the infrastructure.

The project plan serves as a basis for the upcoming documents.

Scope

The scope of this project is limited to the duration of the module [Engineering Project FS2021](#).

References

In the table below you can find the links to the important documents in the repository.

Document	Reference
Project Proposal	<i>Proposal</i>
Risk Analysis	<i>Risk Analysis</i>
Definition of done	<i>Definition of Done</i>
Logging Standards	<i>Logging Standards</i>
Version Control Strategy	<i>Version Control Strategy</i>
Version Control Strategy	<i>Definition of Scrum</i>

Project Overview

Black Fennec is going to be an application that is able to manage unstructured data by interpreting information compositions known to its type system. These interpretations are then visualised. The type system in its nature is a weak typed dynamic object model that can be extended easily. To support specialised use cases and allow rapid development Black Fennec provides an extension api. With the final product one will be capable of visualising and editing any JSON and YAML files in a more productive way. A close integration of git allows for collaboration and data sharing over existing infrastructure.

Purpose & Objectives

The Objective is to have a usable product for data management in our toolset that can be maintained and enhanced for an extended period of time. Therefore the quality of our architecture is of utmost importance.

We will use Black Fennec mainly as a data editing and visualisation tool that allows us to collaborate.

Our hopes are that the project will inspire a community to develop extensions and find use cases that we currently cant even envision. For this we intend to release our source code into the public domain.

To address our personal goals, we intend to utilise and reinforce the knowledge gained during our studies. Simultaneously we hope to learn many new principles and patterns.

Scope of Delivery

The following table defines the scope of delivery of the Black Fennec project.

Documentation	Product
<ul style="list-style-type: none"> • Project Proposal • Project Plan • Risk Analysis • Domain Analysis • Architecture Diagram • Package/Class Design • Code Guidelines • Test Specification/Test Protocol • Protocol notes 	<ul style="list-style-type: none"> • Software build • Source Code • Product Demonstration

Assumptions & Limitations

We expect to accomplish our core objectives before the delivery date. If time permits we will implement further extensions which add additional use cases. These extensions are not part of the engineering project but could be featured in the presentation to show the extensibility and usability of the project.

Project Organisation

Organisational Structure

To manage our project we use Scrum roles and additionally assigned competences to each member. Additionally every member of the team acts as a developer and will contribute code to the project.

Team Members	Competences
Lara Gubler	Scrum Master, Documentation
Simon Kindhauser	Product Owner, Architecture
Leonie Däullary	Development Team, User Experience
Caspar Martens	Development Team, Quality Assurance

Project Manager

The management of the project which includes the scheduling of meeting with stakeholders, and the organisation of development-team internal meetings is in this project done by the role of the Scrum Master which is Lara Gubler.

Project Management

Time Budget

Project duration	14 Weeks
Team members	4 Persons
Working hours per person	120h
Total hours of work	480h
Project start	22. February 2021
Project end	28. May 2021

Time Management

We use Gitlab to track the time estimated and spent time of our work items represented as issues. Fine grained time management will be conducted before each sprint (lasting two weeks).

Phases

The phases that exist in our Project are taken from the Rational Unified Process Framework and consist of the Phases:

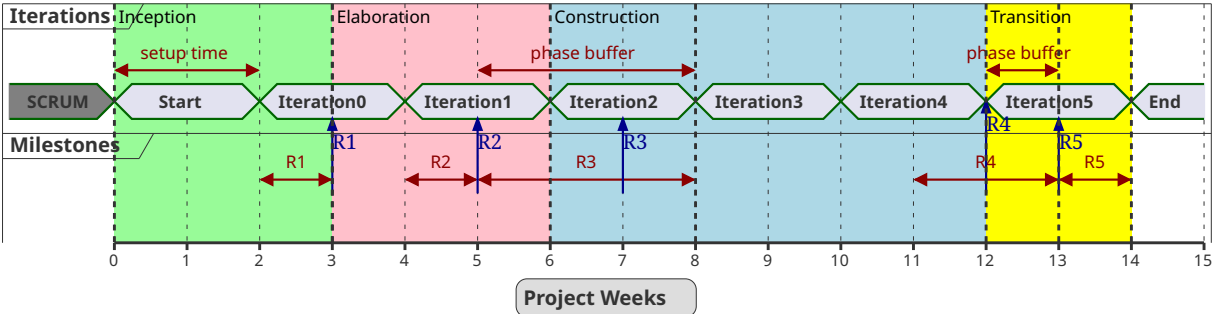
- Inception
- Elaboration
- Construction
- Transition

These Phases do not have a fix duration in our Setup and are evaluated during the planning of the Scrum Sprints. This means no definite duration will be provided here.

Iterations

The iterations during this project are represented by Scrum Sprints. They endure 2 Weeks and are lead by our Product Owner (Simon Kindhauser) and the Scrum Master (Lara Gubler). A more detailed elaboration of our Scrum processes can be found in the chapter Meetings.

Project Timeline



The blue arrows in the milestone section show when Milestones with respecting reviews are planned in our project. As this plan is in the future the red arrows indicate the scope of the time-window in which they should be fulfilled.

During the Elaboration phase a more detailed plan for the Construction phase has been created. It can be found in the Elaboration document.

Milestones

Here in this Document we provide only an overview of the Milestones that exist in our Project in a chronological ordering. At the time this Document is created not all Milestones are already planned to the end. Therefore a link is provided that allows to look at the Milestone in Gitlab which always contains the most timely information. Definite dates and additional deliverables will be defined when appropriate.

Milestone	Link
R0: Project Proposal (Inception)	Milestone: Project Proposal
R1: Project Plan (Inception)	Milestone: Project Plan
R2: Requirements Engineering (Elaboration)	Milestone: Requirements
R3: End of Elaboration & Architecture Prototype (Elaboration)	Milestone: End of Elaboration
R4: Architecture (Construction)	Milestone: Architecture
R5: Q-Review (Transition)	Milestone: Q-Review
Final Submission	Milestone: Final Submission
Project Presentation	Milestone: Project Presentation

Meetings

During our project, various Scrum meetings are held. These always take place on the same day, at the same time, so that all members can plan and prepare for them in advance. The definition and scope of the meetings is documented in our *Definition of Scrum*.

Meetings Timetable

In the following table one can see how we have scheduled the meetings.

Meetings Timetable	Friday 1	Friday 2	Friday 3
08:00 - 09:00	Daily Scrum	Daily Scrum	Daily Scrum
09:00 - 10:00			
10:00 - 11:00	Sprint Review (previous sprint)		Sprint Review
11:00 - 12:00	Milestone Meeting (previous)		Milestone Meeting
12:00 - 13:00			
13:00 - 14:00	Sprint Planning	Daily Scrum	
15:00 - 15:00			
15:00 - 16:00			
16:00 - 17:00		Sprint Retrospective	

Each Sprint consists of two weeks of time but is spread over 3 weeks. A new iteration starts at midday and ends two weeks later before a potential milestone meeting. This accommodates working on weekdays before friday.

The Daily Scrum meetings are held twice a day in the middle week to ensure productivity and resolve potential issues faster.

Protocolling

For each meeting a protocol is created. These record what we discussed in the meeting, what decisions were made and any open issues. They are represented by Gitlab issues which also allow time tracking.

Risk Management

We continuously assess risks and deduce mitigation strategies based on processes described in ISO 27005, specifically assigning the risk one of the following categories:

- reduce
- retain
- avoid
- transfer

Risk Analysis

A list of all identified risks can be found in the document *Risk Analysis*. The most important findings as of the writing of this document are listed below.

Complexity

The mitigation strategy is effective but the issue must be reevaluated regularly to proactively intervene in a timely manner.

User Experience

We dedicate a member of the team to the issue. However, the remaining risk is still significant and we will collectively keep an eye on it.

Quality Assurance

To ensure the desired quality in this project many different standards are enforced. For a detailed account read the document *Quality Assurance*.

The basis of our quality assurance is the use of frequent and builtin meetings. How these are scheduled is described in the *Meetings Timetable*. These meetings are held in accordance with our *Definition of Scrum* and help to ensure quality and efficiency in our code and processes. Additionally, we deploy code reviews and rigorous testing, pair programming and the “gitflow”.

Infrastructure

GitLab

GitLab is a tool which we use for multiple aspects in our project. For example for the management of our source code and documents. We also use it for our version controlling and to plan our project. Our work items are stored in the GitLab repository in the form of issues.

PyCharm

For the integrated development environment (IDE) we use PyCharm from JetBrains. This is a very useful tool for Python programming and includes some useful git functionalities such as commit, push and merge.

PyTest

As our testing framework we use pytest that allows easy to write unit tests but meanwhile also support more complex tests if required. Additionally to test running it is also able to generate a coverage analysis that can be integrated with Gitlab.

PyLint

PyLint is the linter used for our project. It can check for logical errors and formatting. The formatting guidelines used are close to the PEP 8 but differs in minor aspects. A plugin for an easy integration with the IDE PyCharm exists and is used.

Glade

For our graphical user interface (GUI) we use Glade. Glade is a rapid application development tool (RAD) and allows us to quickly and easily develop user interfaces for the GTK toolkit

GTK

GTK is a free and open-source cross-platform widget toolkit. We use it to develop our Black Fennec app.

Elaboration

This document is a starting point to provide a limited overview over the progress that was made during the Elaboration phase. To not only getting a limited overview it is encouraged to take a look at the code that was produced during this phase. The document includes the mention of the created prototypes and the respecting addressed risks and a detailed *Construction Phase Plan*.

Prototype Evaluation

Two prototypes were created to assess risks accurately, test out new technologies and new concepts. The first prototype called tiny fennec due to its inclusion of the final concepts in a simplified manner was created before the project was started to ensure that nothing serious could prevent the project from being successful. The second prototype was produced during the elaboration phase to minimize the remaining risks that were documented in the *Risk Analysis* done in the inception phase.

Tiny Fennec

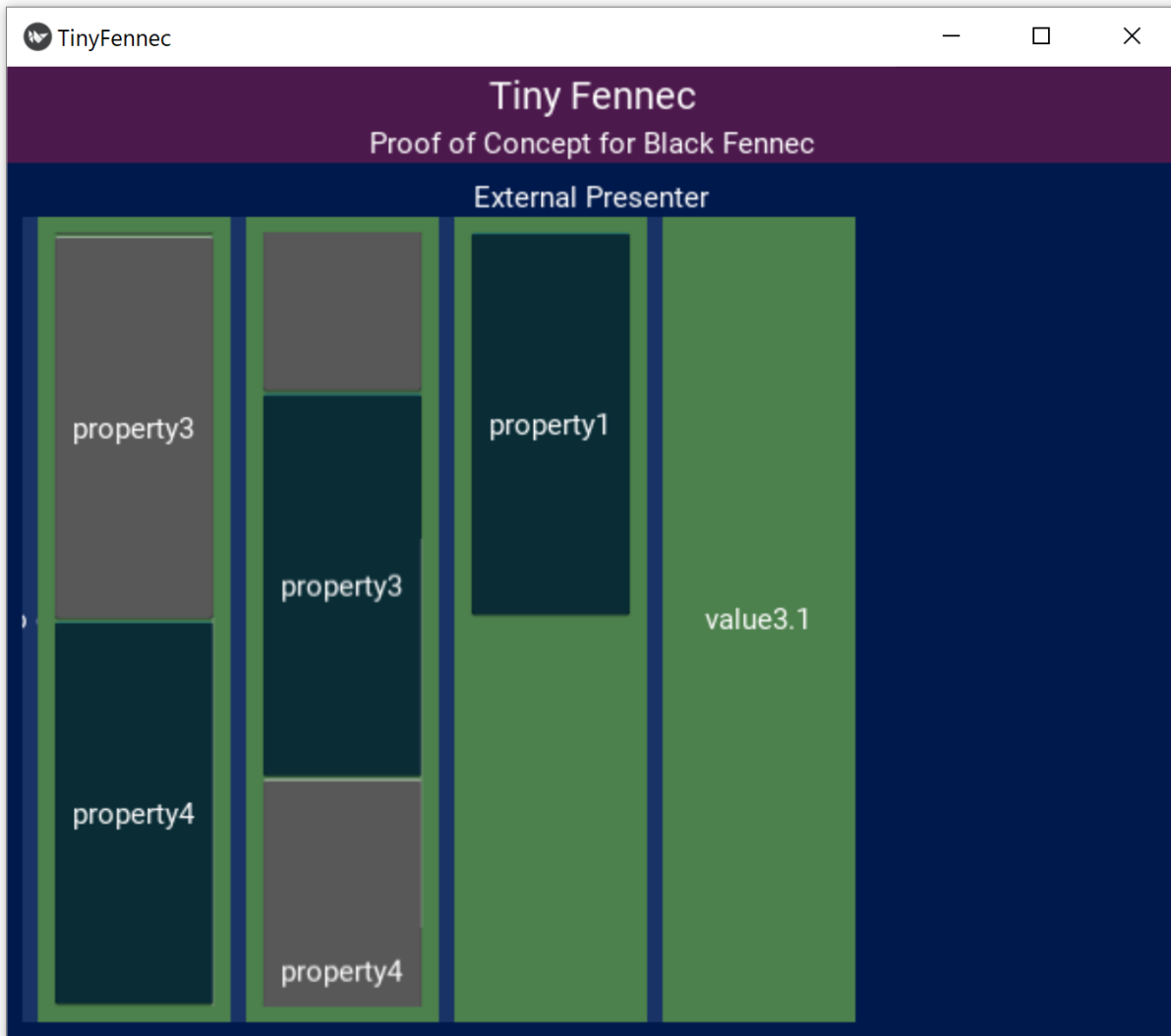
The Tiny Fennec prototype can be found under the following [link](#). In this chapter a preview of the prototype is shown, and the assessed risks are described.

Prototype Preview

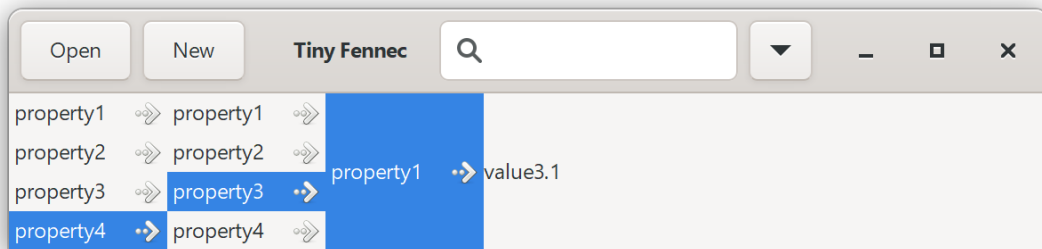
The prototype is not a total spike solution but rather focussed on the feasibility of certain components and concepts. This is why for example the *underlay* is mocked and no *source layer* exists.

The two previews shown represent each a different GUI framework, but both include essentially the same functionality.

Kivy



GTK 3.0



Addressed Risks

Risk	Conclusion
CI/CD	First CI/CD scripts were created, that were able to run the testing framework with a coverage analysis and inclusion of the results in Gitlab. The confidence in the Gitlab infrastructure was established and the pipeline creation syntax mastered on a medium level.
Test- ing	The project includes a test-coverage of 97% and different strategies to ensure testing of components is completely independent were tried. Creation of Doubles and inclusion of dependency injection principle was tried.
Log- ging	Logging was tried out to ensure it functionality, and provisional logging strategy was created.
GUI frame- work	The GUI-frameworks Kivy and GTK version 3 were tried, and the prototype was fully implemented using either framework. The prototype was first created with Kivy and then replaced by GTK 3.0. This also addressed the risk of not being able to exchange the GUI framework. In the end GTK 3.0 was chosen to be used due to the existence of a WYSIWYG editor for the GUI, lesser interference with python, and the possibility of lower coupling of the ViewModel to the View.
User In- ter- face	The concept of a presenter handling the self created widgets of the extensions(types) was tried to ensure the feasibility of this idea. This concept worked out well with either GUI framework.
Ex- ten- sion man- age- ment	Different possibilities of extension management were though through, and the simplest and most powerful solution was the usage of the python extension system with pip and the Python package index (PyPI). For this approach the best practices were evaluated and then implemented in the prototype. The function of it was tested by outsourcing the presenter into a extension and loading it with a Python Wheel.

Black Fennec Architectural Prototype

The Black Fennec architectural prototype can be found under the release tag v0.3.0 and is part of the projects gitlab. In this chapter a preview of the prototype is shown, and the assessed risks are described.

Prototype Preview

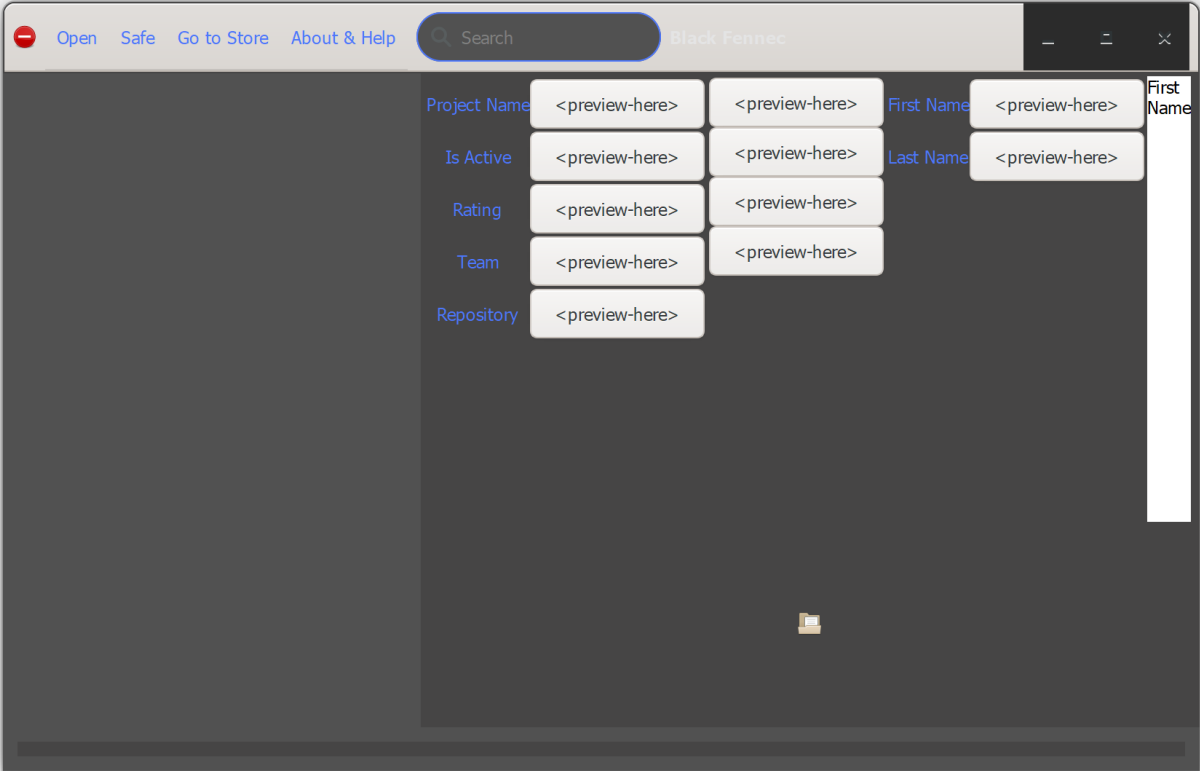
The Black Fennec architectural prototype is a spike solution and is able to open and parse JSON files, and show their data. In the prototype the json types are being handled and thus any JSON file can be viewed. Under the subsection *Main Window* an example file representing project metadata can be seen as it is presented in the prototype. The prototype includes most of the required domain concepts, and is written to be extended easily with additional features.

Loading Screen



The loading screen of the prototype is to be extended to contain status updates and thus until now the progress bar remains to be stationary.

Main Window



As this is a prototype, some of the already displayed functionalities such as saving a file, accessing the extension store

or showing a help are not implemented. Also the concept of previews of data are to be implemented but are planned in the construction phase. Because of this items of a list are not identifiable. One there can imagine a compact view of the most important value of the type contained in the list.

Addressed Risks

All risk included in this table correspond to a heading in the *Risk Analysis* document.

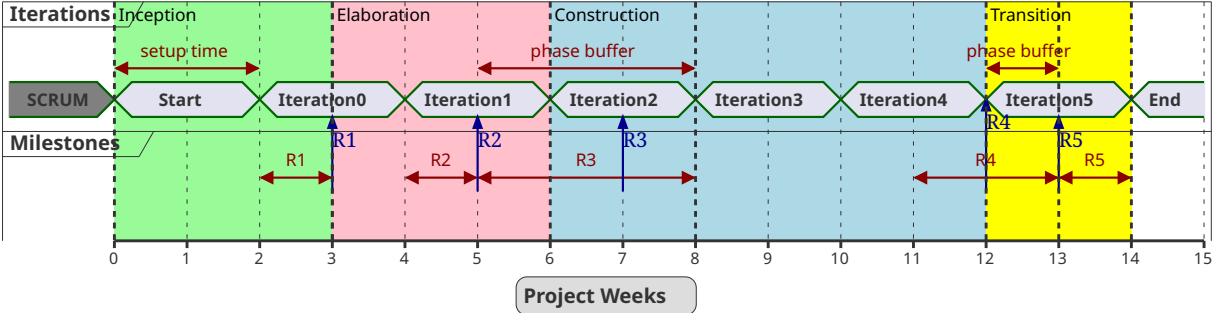
Risk	Conclusion
<i>Reaching the Complexity Ceiling</i>	In order to mitigate and minimize this risk a great deal of time was spent to refine and refactor concepts in the core such as the auctioneer and navigation service. Thanks to this measure we are confident that the complexity encountered during this project in manageable. But still a complete mitigation of this risk can never be claimed, and thus we have to keep this risk in mind.
<i>Changing Requirements</i>	During the elaboration phase the requirements of our project have been concretised, and thanks to this we can conclude that this already very low risk has further been lowered.
<i>Documentation Tools</i>	The prototype features an extensive documentation in the form of docstrings. Contrary to popular belief that docstrings are not maintainable and do not add worthwhile value, our team decided to use docstrings especially to ensure that external extension developers can understand the underlying code of the Black Fennec core with lower time investments. We deliberately refrain from documenting our tests and doubles, but core code is to be documented at all times. We believe our documentation tools to be adequate and thus see this risk as mitigated.
<i>Development Tools</i>	The members of the project have set up their IDEs preceding the start of the elaboration phase as a part of the inception phase. Now up to this point all members have made contact with all the development tools and any occurring problems were successfully solved.
<i>Low Software Quality</i>	Our Prototype adheres strictly to our quality standards, and with the inclusion of a code quality rating with pylint this risk has been mitigated. Up to this point all members of the team have made contact with the quality standards and know how to adhere to them.
<i>Bad User Experience</i>	This long-term risk still exists for our project, but with the assignment of one team member that is has the competence to ensure best possible user experience, at least a bit of this risk has been mitigated.

Retained Risks

Risk	Conclusion
<i>Third Party Component</i>	As already stated in the <i>Risk Analysis</i> this risk was retained an remains in this state, as it is not feasible in the scope of this project to provide a mitigation for this risk.

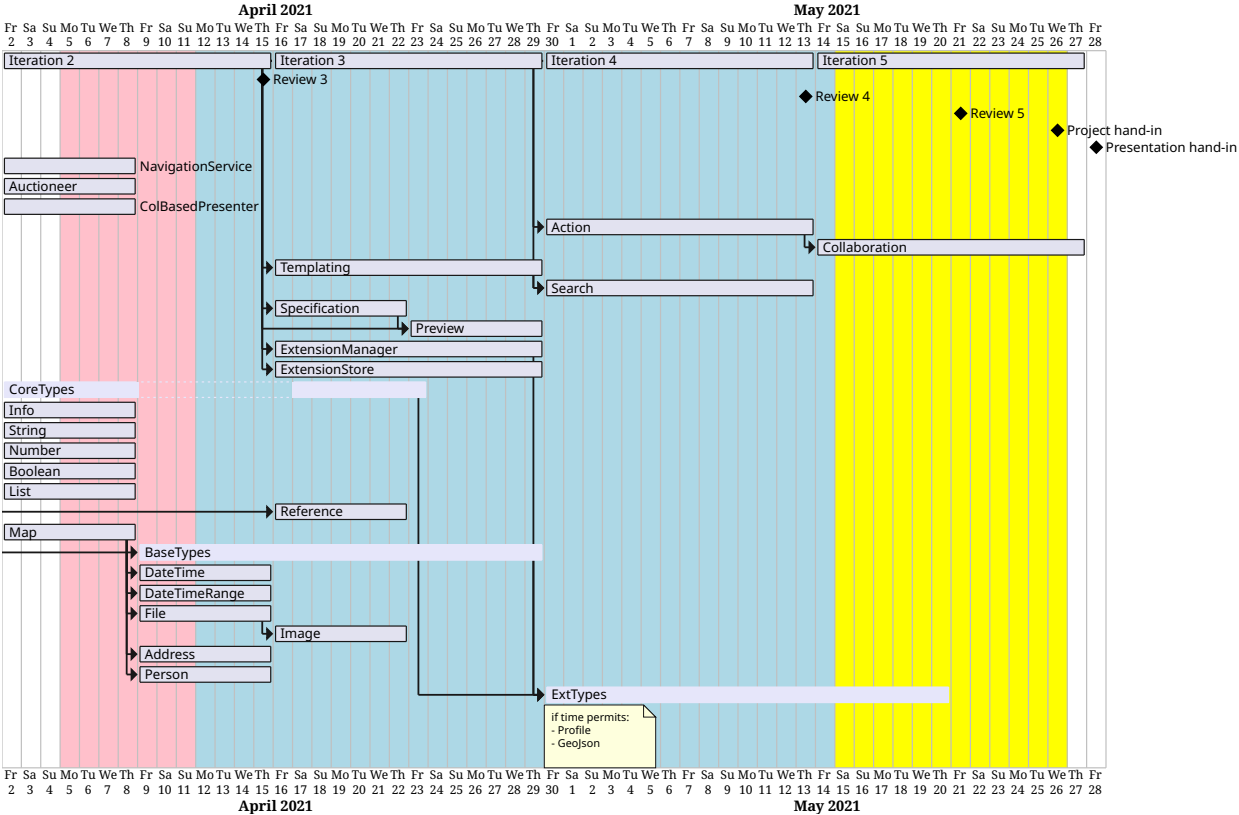
Construction Phase Plan

The overall timeline of our project was created during the inception phase and also is included in the *Plan*. During the course of the Elaboration phase it was adjusted slightly but besides the prolongation of the elaboration phase the plan remained identical.



The Construction phase spanning at least the two next iterations is planned more detailed in the following diagram. Additionally to this plan, a look into the *issue board* can be worthwhile to get a picture of the current and planned processes.

The timeline we provide is a gantt diagram to visualise dependencies of certain component of our system and provides a detailed planning overview. This plan is just a provisional aid for the creation of stories and work-items during the sprint planning meetings, and is thus subject to change.



Refinement

After each sprint, a review meeting was held to discuss the current status of the project. The checklist of the individual milestones was discussed and it was ensured that all items on the list were fulfilled. During the meetings, we received some suggestions for improvement from our supervisor, which we implemented during the course of the project.

This document provides an overview of all these improvement suggestions and how they were implemented in our project.

Project Plan

Improvements

Role Project Lead / Project Manager must be added in the documentation The description of this role was added in the document project plan under *project manager*.

Rough plan of the project must be created and added in the documentation We created a project timeline in the document *project timeline*

Add the time needed to respond to the risk to the risk analysis and the rough plan of the project The time needed to respond to a risk is included in the *project timeline* by the use of phase buffers but do not address specific risk but rather correspond with when risks are most likely to occur.

Requirements Engineering & Domain Analysis

Improvements

Use the personas in the use-cases We did this by relating the user stories to specific personas. An example of this can be found in the following issue (issue does not exist anymore).

Add sketches from extensions as soon as they exist These sketches were added in the form of wireframes to the domain model document.

Architecture Prototype & End of Elaboration

Improvements

It is hard to make a connection between the architecture documentation and our structure in the code. We should therefore consider either adapting the documentation or the folder structure With a major refactoring of the structure we ensured that the architecture documentation matches the structure of the code. For more details on this refactoring one can look at following issue (issue does not exist anymore).

Software Architecture

Improvements

Consider adding a performance test. E.g. insertion of large files Performance tests were added at the two identified bottlenecks that our application has and are now included in the system tests performed before each release. For more information look at the *performance scenario* chapter.

The document contains only static views of the system. Consider adding a dynamic view of the system Multiple dynamic views in the form of three sequential diagrams were added to the documentation.

Quality Ensurance Measurements & Code Quality

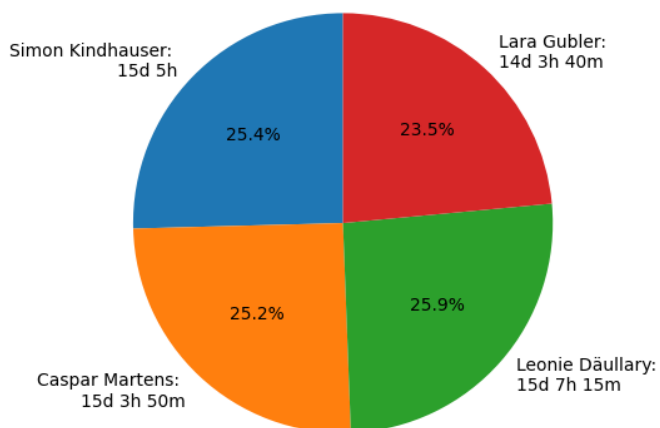
Improvements

Consider adding usability tests A *usability study* was conducted with the participation of four completely uninvolved people, which we selected on the basis of the *personas* we created.

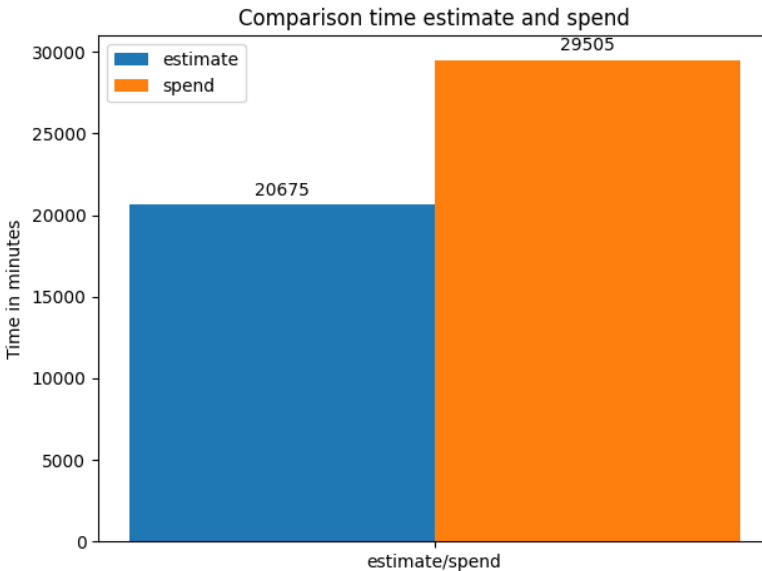
Time Tracking

In this document automatically generated charts based on the Gitlab time tracking features that can be recorded on issues and merge requests will be featured. This is automatically initiated through a CI-Pipeline which with the help of python and the module matplotlib generates charts that contain the data collected via the tool gtt from the Gitlab API. This Pipeline is a Work in Progress, which is why in this version a manually executed python script generated the chart below.

Time spent per Team-Member



Comparison of estimate to time spent



Reflection

The engineering project is coming to an end. We experienced this project as a tightly coupled team, which is why, besides duplicate «code», we decided to write the reflection on our work in a team effort. We have learned many new things, and reinforced many things that we knew only from a theoretical perspective.

Two modules that gave us the theoretical basis on this project were Software Engineering 1 & 2, in which we did not yet have a real practical experience to test the learned techniques. One of the things learned in SE1, the usage of git, was not totally clear for all of us at the beginning of the project, but over a short period of time, became completely logical. But much more important were the project management practices such as SCRUM/agile, which we strictly followed during the whole project. We noticed that planning a project is a lot of work, and comes with a certain overhead. The time spent planning and creating documentation of the planning took much more time than anticipated. Nearly a third of the time we would have usually spent programming was invested into planning and reviewing our processes. But nevertheless the planning provided us with a guidance that was very useful to ensure the quality and timeliness of our project.

Technologically we have also learned many new things. For example the programming language python was newly learned by three of the four team members. This goes from simple things, for example structural typing to more advanced disciplines such as testing, linting, CI-CD, etc. Another totally new technology learned for all of us was the GUI-Framework GTK, with which we built our application. Regarding the documentation, which we decided to set up on our own, was also managed with a new tool called Sphinx. Eventhough we did not know how to use these technologies at the beginning, we began to understand them without much effort, as the concepts behind them did not differ significantly from other programming languages, frameworks or documentation engines we already knew. This leaves us with the conclusion, that as long as you understand the core concepts learning new things based on these concepts is quite easy.

Because we are very pleased with our work and the code quality produced we are looking forward to work with this project as a foundation for future projects building up upon Black Fenec. We plan to release this project into open source soon, and will then work on further improvements and features in our free time or maybe even in a semester project or bachelor thesis.

We had a lot of fun during this project and learned many new things, and if we had to do it all over again, the path we

would take would only differ marginally.

Study Project

The *study project* is a semester long project that is carried out by a group of two students. Accumulated about 480 hours were invested into the development of Black Fennec. In contrast to the preceding *engineering project* only two (compared to four) members had each double the time on the project resulting in the same amount of accumulated work hours.

This project, a student project of the Eastern Switzerland University of Applied Sciences (OST), was supervised by Prof. Mirko Stocker.

The *project plan* provides a rough overview of our project, the project organization, and the management processes. As this project was a continuation of the *engineering project* and only two people developed it, it required significantly less communication overhead. In the previous project the Scrum process was identified as too much overhead and was as such not very suitable for a project of this size. Most of our planning was done inside of the issue board of our projects repository. Formally documenting this process here would have been - in our eyes - of lesser value to the intended target audience (future developers) than extending the documentation in other more valuable sections.

To provide an overview of what was achieved over the course of this project an *Increment* in the form of a change overview from the version 0.6 - which was the version from before this project - and the final release was created.

Additionally to the aforementioned subsections this section also contains the projects *time tracking* and a *reflection statement* that was written at the end of the project.

An overview of all the mentioned subsections can be found in the following table of contents:

Proposal

Supervisor

This student research project is supervised by Prof. Mirko Stocker, mirko.stocker@ost.ch. This assignment is based on the guidelines for bachelor and student theses of the computer science program.

Students

This work is carried out as a student research project at the department of computer science by the following students:

- Caspar Martens
- Simon Kindhauser

Introduction

The idea for Black Fennec arose from the need for a tool that can assist in the collection of any data. Black Fennec is positioned between Microsoft Word and Microsoft Access and is intended to compete with their use for the purpose of data collection. It can be understood as a just-in-time form generator or data store.

With the semi-structured approach, we are able to offer manipulation and display possibilities that would not be possible with continuous text or classic databases.

The current version supports basic functionalities like editing and creating datasets. These data sets can be processed in JSON format. Great importance was attached to the extensibility and so a plug-in system allows the addition and removal of data types at runtime.

Goals

The goal of the work is primarily the publication of Black Fennec as a product. For this purpose the project shall be brought into a stable and maintainable state, the architecture and the code shall be critically reviewed and improved and a number of not yet existing functionalities shall be added.

In the spirit of agile project planning, the following epics are part of the backlog

- Reduction of technical debt
 - Architecture refactoring
 - JsonPointer
 - File caching and saving
 - Program settings
 - Plugin system
- New features
 - Stable templating API
 - YAML support
 - Actions
 - GTK4
 - JsonPath
 - XML support
 - Collaboration
 - Plugins
 - * Base
 - * OSINT
 - * GeoJson
 - * Graph Presenter

The exact features are defined by the team during the project work. The work is planned as four three-week sprints, which always produce a deliverable product at the end. In the last sprint we focus on the public publication of the project (Gitlab/Github, publication on Flathub, website).

Realization

The course work is about applying the knowledge you have learned in the various modules of OST to a project. In particular, your software engineering skills will be required. You will be expected to apply this knowledge and use methods such as unit testing, clean code, SCM and continuous integration whenever possible.

Meetings with the supervisor or client will be held once or twice a week, the exact rotation will be mutually agreed upon at the beginning of the work. Additional meetings are to be arranged by the student as needed.

All meetings, except the kick-off meeting, are to be attended by the students with an agenda. (What was done? What was accomplished? What was not? What questions do we have?). As a rule, a meeting should not exceed one hour. The agenda should be sent to the meeting participants at least half a working day in advance. The results of the meetings (i.e. the decisions made) are to be recorded by the students and then sent to the supervisor or stored in a defined place (e.g. wiki).

Tools

Unless specified in the assignment, students are responsible for selecting their own tools, libraries, frameworks, SaaS offerings, etc. If desired, a virtual machine hosted at OST can be provided.

Timeline

Date	Event
19.09.2022	Start of the study work, issue of the assignment by the supervisor. Templates as well as detailed instructions regarding documentation are available on teams.
19.12.2022	Students enter the abstract in the online tool http://abstract.rj.ost.ch and release the abstract to their supervisor for review.
22.12.2022	The supervisor releases the document with the correct and complete abstract to the program secretary for further processing.
22.12.2022	Submit report to supervisor and upload all documents to https://avt.i.ost.ch by 17:00.

Evaluation

Successful coursework counts for 8 ECTS points per student. For 1 ECTS point is a work performance of 30 hours. The responsible lecturer is responsible for the assessment.

Aspect	Weight
Organisation, realization	20 %
Formal quality of the report	20 %
Analysis, design and evaluation	20 %
Technical implementation	40 %

The grade is announced to the students via unterricht.rj.ost.ch. The lecturer can/may indicate whether the work is considered pass/fail, he/she may also give advance warning if there is a risk of an unsatisfactory grade.

Abstract

Introduction

Black Fennec is a simple and easy-to-use application for viewing and editing semi-structured data like JSON in a visually interpreted form. The target audience is people who come in contact with such data regularly and want to collect and manage it in an accessible way. It is positioned between Microsoft Word and Access and can be understood as a data editor.

It was originally developed by Lara Gubler, Leonie Däullary, Simon Kindhauser, and Caspar Martens as an engineering project with Thomas Kälin as supervisor. Building on the previous work, the goal of this project was to release Black Fennec as a stable and mature product with major quality-of-life improvements that potential users are keen to use.

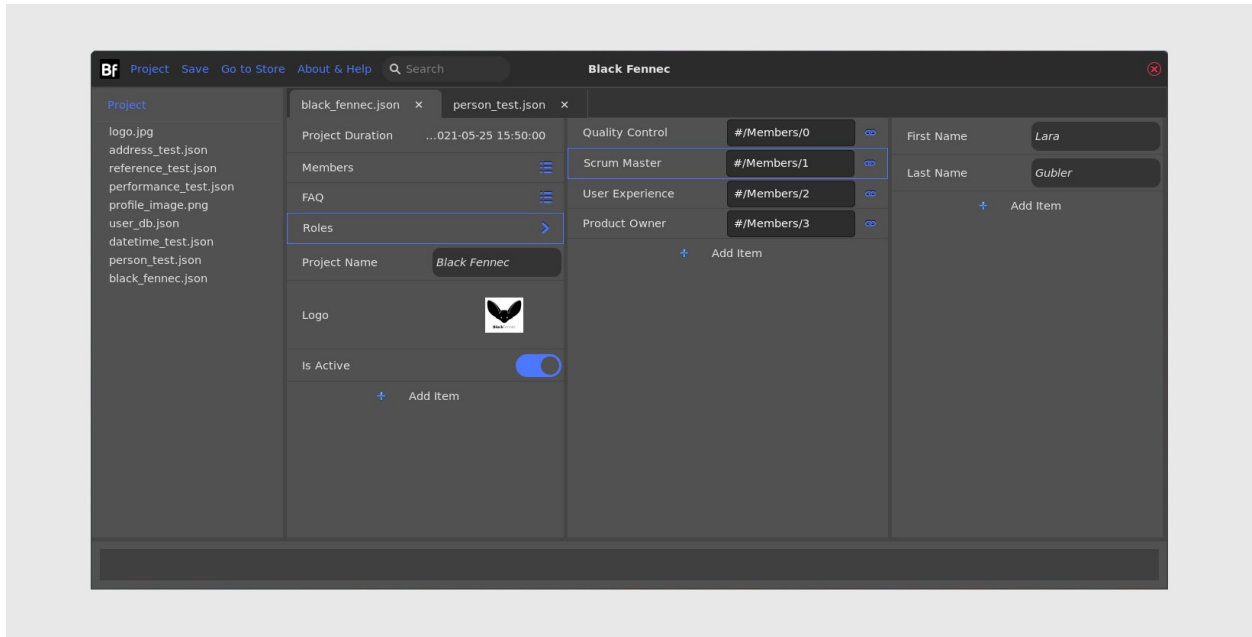


Fig. 1: State of the application before the project at version 0.6.

Approach / Technology

Improving the stability of the application included major refactorings to many components of the application. Architecture-wise, some of the most significant changes were made to the document subsystem, extension handling, and the type system. This was made possible by an extensive and largely pre-existing automated test suite.

The test suite itself also received a significant overhaul. The pre-existing suite, which was using the python unittest package, was largely extended and simplified through the use of pytest. To acquire the necessary skills the developers took part in a multi-day workshop organized by members of the pytest community. Currently, a thousand tests cover around 98% of the code base (branch coverage).

Thanks to the migration to GTK 4 and the use of libadwaita, the application integrates seamlessly into the GNOME desktop environment. It also improves maintainability by providing a large set of components that are responsive and user-friendly by design. We leveraged these technologies in an effort to improve the user experience of the application. Amongst other things, this included the addition of application feedback and better error handling.

With the introduction of actions to the extension ecosystem, we have also created a new way of interacting with the application. Actions allow the execution of previously defined operations that operates on a selected data structure. Overall, the extension API has matured and now offers a smorgasbord of extension points including actions, mime types, and more.

As a means of distribution, the application is now available as a flatpak package on Flathub. This allows for integrated installation and updates on all major Linux distributions through their native software center. Another advantage of flatpak is that it provides a sandboxed environment that adds another layer of security and protects our users from malicious intent.

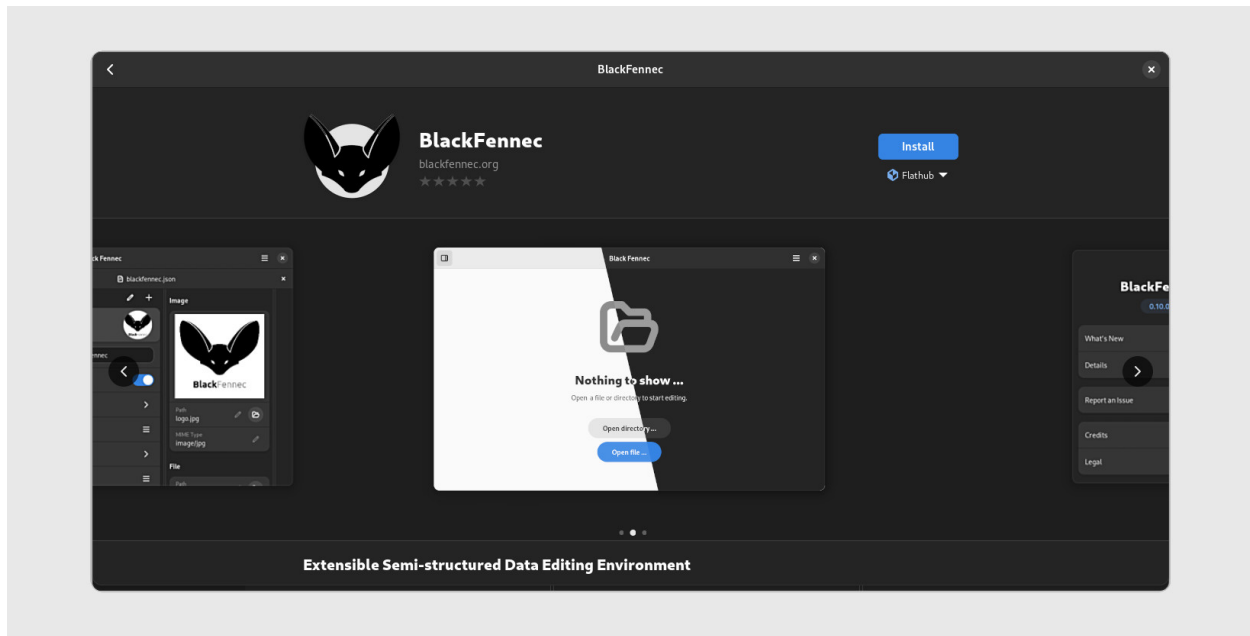


Fig. 2: Gnome software store with Black Fennec. The application is available via Flathub and thus any Linux distribution.

Result

We are happy to announce that Black Fennec is now available as a stable and mature product. The application can be downloaded from Flathub and installed on virtually any Linux distribution. Complementary to the release of the application, we also published a simplistic website with all relevant information including a short description, a screenshot, and links to other relevant resources.

The website is available at <https://blackfennec.org>

Project Plan

The project plan shows how the project was managed and planned. It has the goal to provide means to evaluate the project management and planning process which was not done formally in the form of a concrete document.

The *lessons learned* from the preceding *engineering project* <engineering_project> influenced this project greatly, as the last project had too much overhead for what was done. Contrary to this project the overhead was expected and intended by the scope of the project. From the taken lessons learned we decided to reduce the overhead by slimming it down, and increasing the duration of a sprint.

The planning was done mainly on the *issue board* in the Gitlab instance. We used labels and milestones to further structure the issues.

We created a *milestone* for each sprint and assigned the issues to the sprint they were planned for. We decided on sprints of the length of 3 weeks, as this was deemed the optimal length for the project, which turned out to be true.

In the initial planning the backlog was populated with epics that were defined in the *project proposal* as the scope of the project. The assignment of these epics to specific sprints was re-evaluated on a regular basis and the epics were split into smaller issues or sub-tasks if necessary. The issues were then assigned to the sprint they were planned for and to the developer responsible for the implementation.

Additionally to the issues in the form of epics/user stories, we also created issues for tasks or literal issues that were

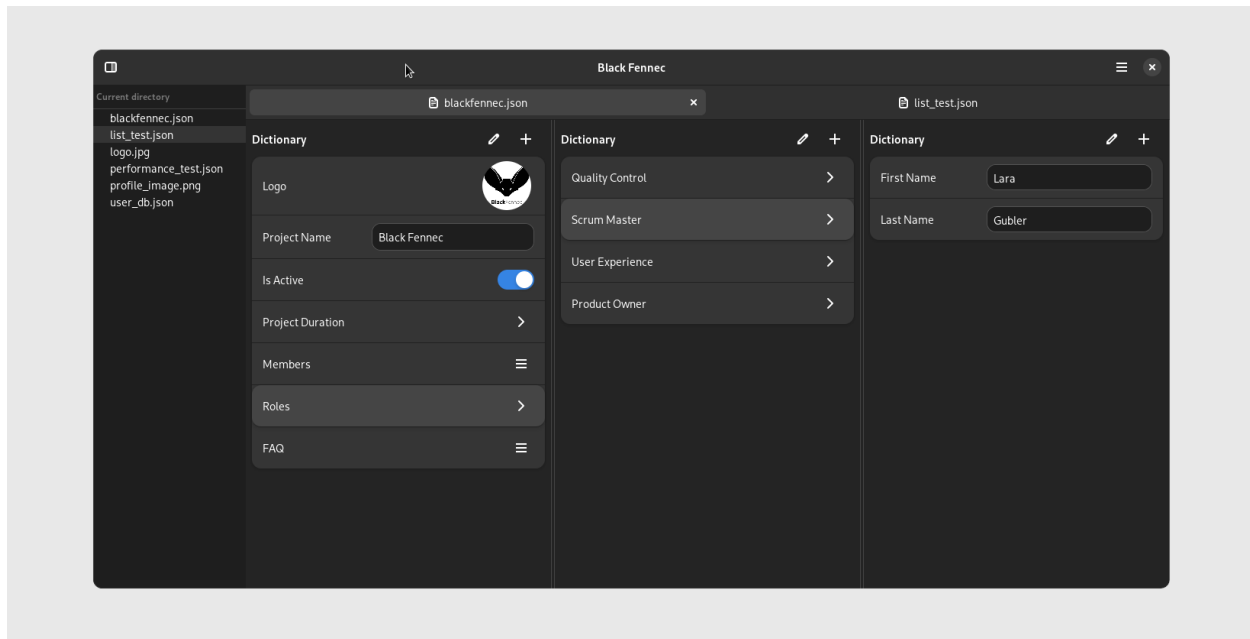


Fig. 3: Final release of Black Fennec at version 1.0.

encountered during the course of the project. These issues then functioned as fillers in the sprints where no additional epic would fit but still some time was left.

Increment

The increment between version 0.6 and 1.0 represents the changes and improvements made to the codebase during the study project. To gain an overview over the changes, multiple resources exist as a reference. In the [milestones](#) one can see which issues were associated with which release. The [changelog](#) contains a list of all changes made to the codebase. The [commit history](#) shows all commits made to the codebase. The [merge request](#) between the two versions shows all changes made to the codebase.

As these resources are not easily accessible to gain an overview, the following visualisations will give a short overview over the changes made to the codebase.

The diagram illustrates the significant changes to the codebase, which were not only comprised of additions but also numerous refactorings. However, it should be noted that the decrease in lines of code (LOC) may be partially attributed to the transition from a monolithic repository to a multi-repository structure, which involved the removal of the core and base extensions.

The above figure presents the proportion of code written by each author in the codebase. It is evident that all of the code was created by the members of the two projects, with the leading author accounting for more than 50% of the code in the *engineering project*. However, it is important to acknowledge that this data does not necessarily reflect the complete contributions of each author to the project, as it only considers the lines of code written and does not take into account other forms of contribution such as documentation, code review, and pair programming. During the course of this project, the difference in the amount of code written by each member was reduced as a result of significant refactoring by the two project members. This refactoring is also evident when examining the LOC contributed by the members of the *engineering project* that were no longer part of the succeeding project.

Here in this figure it shows the changes made to the codebase over time, with the lines of code written by each author stacked on top of each other. It is clear from this figure that the codebase has undergone significant changes, with both additions and deletions of code over time. The decrease in the overall number of lines of code can partially be attributed

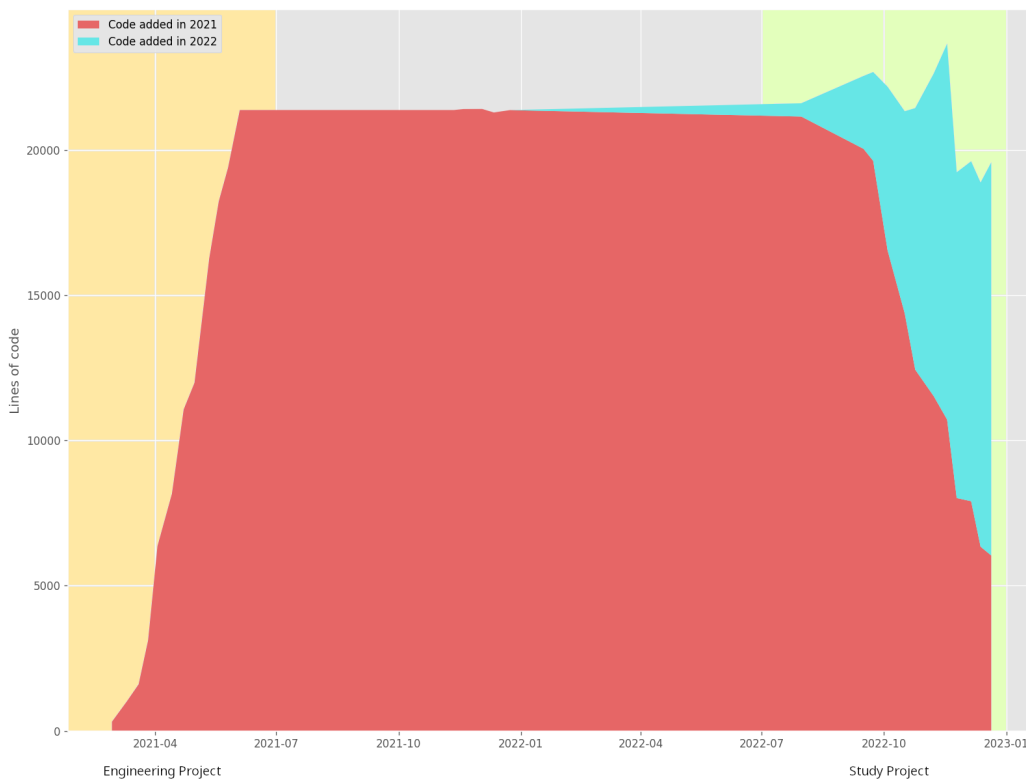


Fig. 4: Lines of code split per year showing the changes made to the codebase.

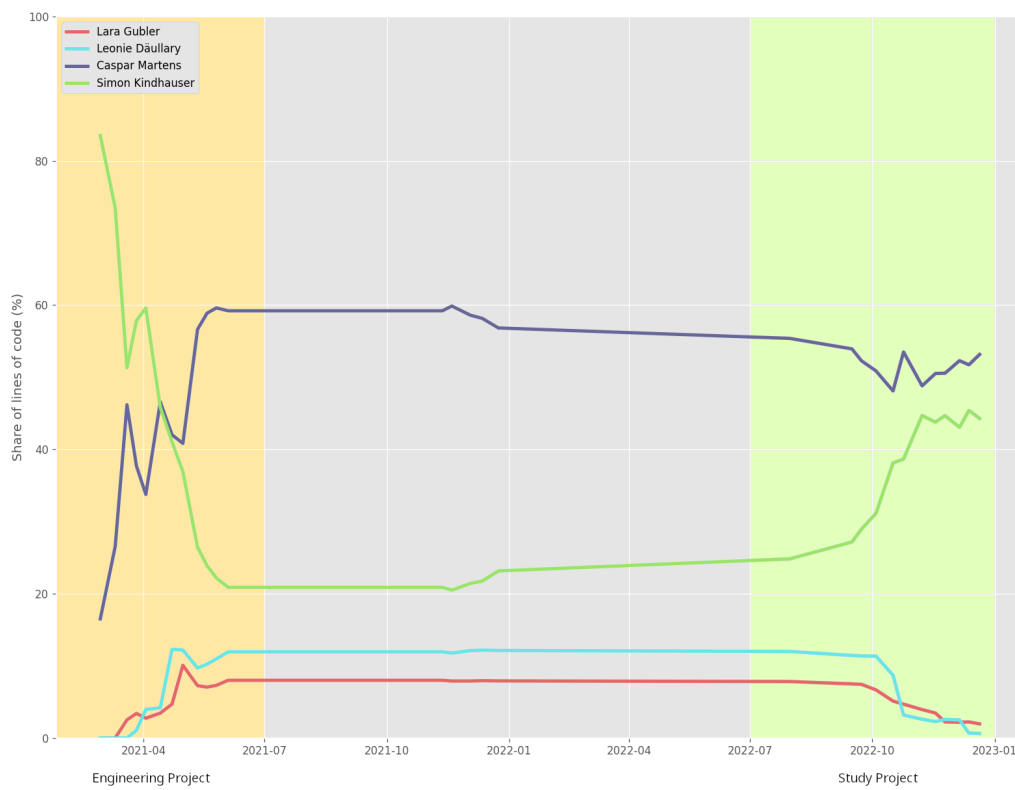


Fig. 5: Percentage of code written per author.

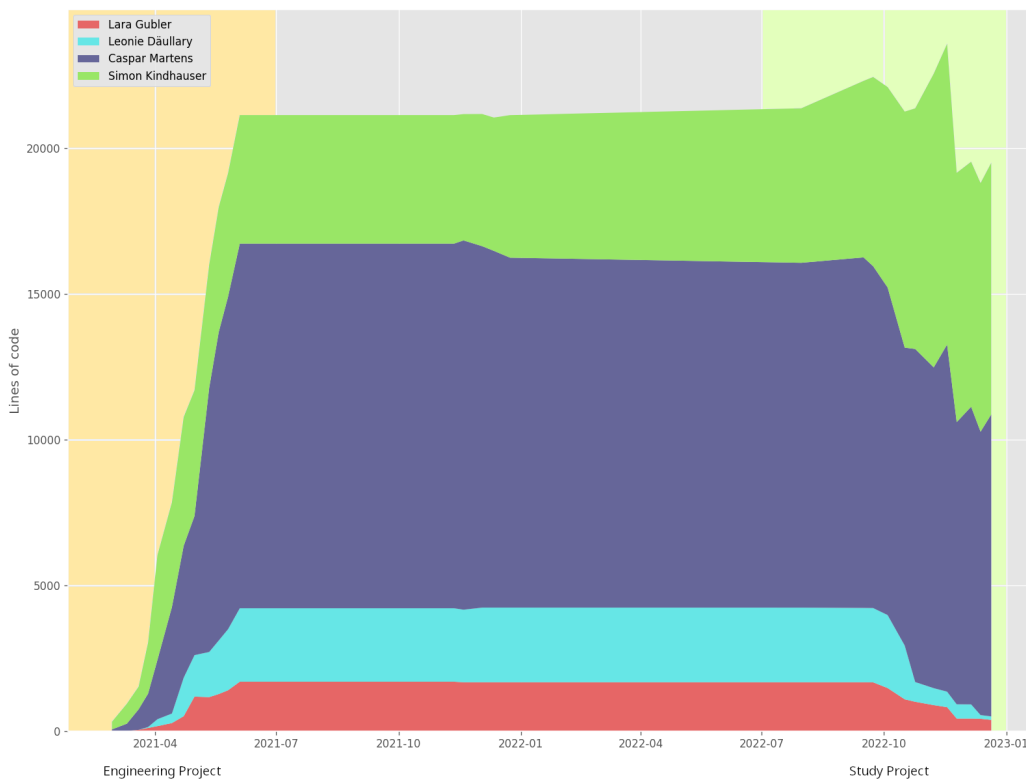


Fig. 6: Lines of code split per author showing the changes made to the codebase.

to the transition from a monolithic repository to a multi-repository structure, which involved the removal of the core and base extensions. Another factor for the decrease in lines of code is due to refactorings and other code cleanups performed by the authors.

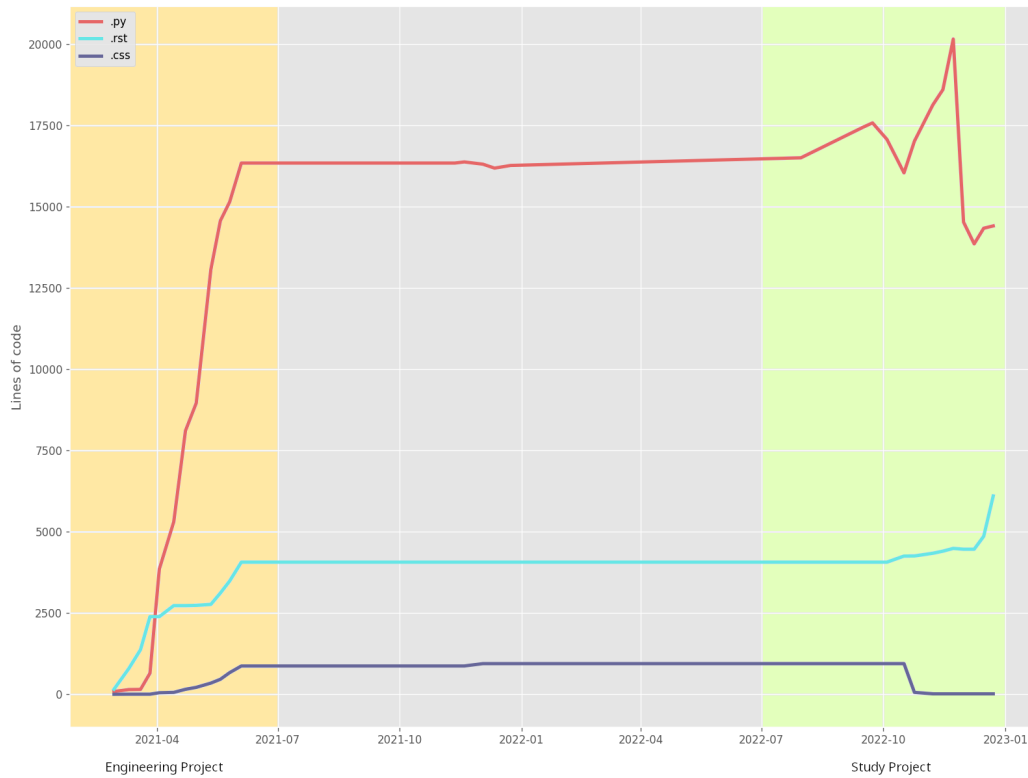


Fig. 7: Lines of code split per extension showing the changes made to the codebase.

The figure above illustrates the changes made to the codebase over time, with the lines of code written for each file extension shown separately. A notable change is the decrease in the number of lines of code written in CSS, depicted by the dark blue line on the graph. This decrease is a result of the removal of the CSS for the Black Fennec UI, which was replaced with the use of libadwaita and GTK4, which provide sufficient styling when used in combination. It is also noteworthy that there has been an increase in the number of lines of RST content towards the end of the *study project*. RST is the language used by Sphinx, the documentation framework employed in this project. Compared with the additions made in the engineering project only a slight increase can be seen, as huge parts of the documentation were only rewritten and thus did not increase the amount of lines of RST significantly.

Time Tracking

In this project, we divided our time tracking into three categories:

- Project Management
- Documentation
- Programming and Testing.

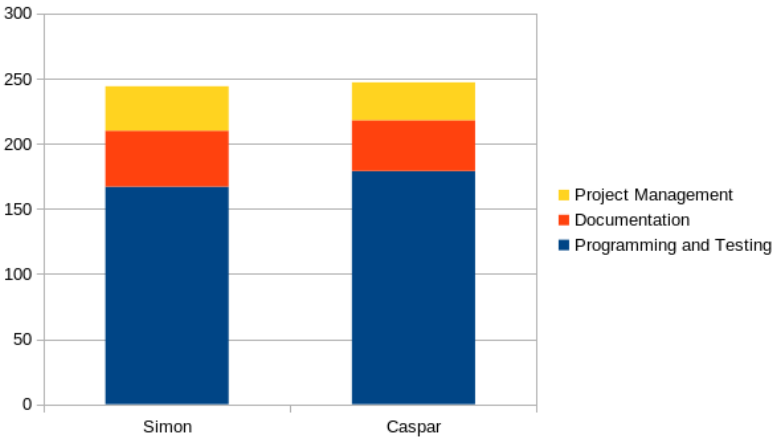
The Project Management category encompassed weekly meetings as well as discussions about architectural changes and potential technologies and libraries. The time allocated to this task remained fairly consistent throughout the project and did not exhibit significant fluctuations.

The Documentation task required the majority of its time towards the end of the project. While we began updating the architectural and domain model sections from the outset, many parts were revised and finalized towards the end to prepare the application for release. In addition, the separation of the engineering project and the study project, as well as the creation of artifacts for the study project, were mainly completed towards the end of the project.

As with the project management the programming and testing was spread widely and only stagnated to the end of the project, where most of the time then went into the improvement of the documentation.

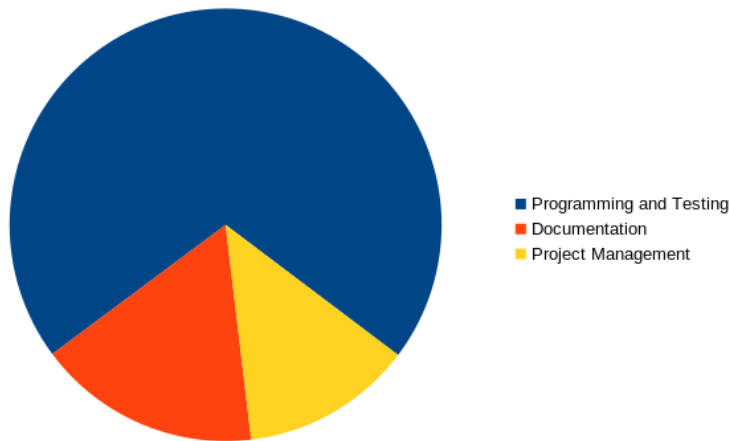
The following two diagrams illustrate the amount of time spent on each task, as well as the task that consumed the most time. The first diagram shows the time spent on each task by individual, as well as the total time spent by each person. The second diagram compares the overall time spent on each task during the project.¹

Time spent per Team-Member on tasks



¹ The pyramidal configuration of the diagram serves to visually underscore the precision with which it reflects the allocation of time among the diverse tasks at hand, much to our delight.

Time spent on tasks in total



Reflection

Black Fennec will soon be released and it is time to look back and reflect. Our hard work has payed off and we are pleased with the result and that we have achieved our goals.

In the previous project, we found that using a full-fledged Scrum approach with two-week sprints that we only worked on one day per week was excessive and had a lot of unnecessary overhead. As a result, we streamlined our project management process and extended our sprints to three weeks. This change worked out exceptionally well and allowed us to accomplish significantly more during this project compared to the previous one.

At the start of the project, we had plenty of time to dedicate to Black Fennec and were determined to stick to our project plan by focusing on maintaining the scope rather than the timeline. However, as we prepared for a presentation about the project in the “Project Management and Quality Management” module, we realized that we had been spending too much time on Black Fennec and decided to reduce our efforts. We adjusted the timeline instead of the scope and made up for the overtime. At the same time, we also had many other modules to attend to, many of which were evaluated during the semester. We significantly underestimated the amount of time they would require.

Fortunately, as we gained familiarity with the technologies and codebase, the time needed to implement changes or features decreased over the course of the project. This allowed us to maintain the original scope and timeline without needing to adjust either.

We believe that the time required to make changes to the codebase is highly correlated with the maintainability of our code. As we progressed with refactoring, which we did to a large extent at the beginning of the project, it also influenced the time needed. These refactoring efforts would not have been possible without our rigorous testing.

That’s why we place a high value on test coverage. It required discipline to write test cases for every path in every function we wrote or modified, but we motivated each other through code reviews and a definition of done that emphasized good coverage. The rewards of this effort were well worth it, as effective and efficient refactoring allowed us to make significant improvements to large parts of the application. Without these efforts, it would have been impossible to push the boundaries as far as we did.

Design Decisions

This section of the documentation serves as an overview of the architecture and package-/class-design. This document serves only as an overview. More detailed information can be found in the documents referenced in this overview.

Physical Architecture

Black Fennec is a single user application running on a single computer and does not interact with the network necessarily. The main engineering challenges in the project stem from the complexity and some of the required algorithms to implement the functionality.

Logical Architecture

Black Fennec is best described as a set of services and interfaces that serve different stakeholders and allows communication between them. The main packages are described in the Architecture documentation and a detailed overview over the most important components are documented in the domain model. Each loosely coupled component solves a different problem. For example, the *Structure* abstracts data access, filtering and implements recursion via *References*.

Application and Programming Interfaces

The packages have been designed with low coupling and high cohesion in mind and exist largely independent from each other. This was necessary to manage the complexity of the project. The interfaces are weakly typed and rigorously tested as this is the pythonic way.

Persistence

Since we do not have a backend or database, persistence was not a real problem which needed solving in our case. Nevertheless, serialisation of user data was implemented with the visitor pattern giving us great flexibility in the output format (JSON, YAML etc).

Package Design

We deploy a great many design patterns to manage the complexity of this project; MVVM, Abstract and Simple Factory, Visitor to name the most prominent ones. Listing them all here would be besides the point.

User Experience

To ensure the user experience, we created a special role in the team. We started early on with design concepts and integrated UX requirements with the help of *Matthew G. Gritton*. Additionally we created a usability study which is to be done for new feature included in a release. The template can be found [here](#).

Architecture Highlights

The main feat of the architecture is its capability to hide enormous complexity in simple to use services. Many of our components require significant time to understand - not because they are poorly implemented but due to the sheer complexity of the problem they are solving. But even so, using them is as simple as it gets. This allowed us to work in parallel at different aspects of the system and resulted in near seamless integrations and extensibility.

Possible Extensions

Although Black Fennec is a very viable product, we have yet to deliver some of the features teased in the project proposal. The most prominent of them are *actions*. We recently decided that this feature could not be implemented within the time constraints of the engineering project. However, it is perfectly feasible to add *actions* in the future. As a matter of fact, we plan to do so.

Furthermore, Black Fennec is designed to be extended by *extensions*. This allows users as well as developers to implement new functionality as they see fit. The possibilities are quite literally endless.

Performance Scenario

This isn't much a concern for us as Black Fennec is a single user desktop application. Our project is never used by more than one user. Anyone who wants to use Black Fennec can install the application locally on the machine and work with it. The performance bottlenecks that we see are mainly related to file size and the amount of installed extensions.

System Test

Which is why we have a dedicated system test that ensures that large files (1 MB/ 100'000 lines) can be opened in under one second. This is tested each release cycle, to ensure nothing has broken this requirement.

To ensure that the second bottleneck that we have identified, namely the installation of many type plugins, each bidding on every viewed structure. For this a special extension, loading the same bidder a thousand times while opening a large files (1 MB/ 100'000 lines) was created.

Risk Analysis

This document contains a list of all identified technical risks to the Black Fennec engineering project. It is updated on a need be basis.

Reaching the Complexity Ceiling

Damage	Very High
Probability	1.0
Danger	Very High

The project is too complicated or complex for developers to effectively make changes or add features.

Mitigation

Strategy	Reduce
Effectiveness	0.5
Remaining Risk	Medium

Managing complexity effectively is very hard. We must deploy many techniques and tools to mitigate this risk effectively. We will invest heavily into the architecture of the system and additionally strive for flexibility for when simpler solution arise they may be implemented. Flexibility through refactoring is aided by unit tests, giving the developers confidence in rapid changes.

The remaining risk must be watched carefully and further analysis will take place before the issue raises to dangerous levels.

Changing Requirements

Damage	Medium
Probability	0.6
Danger	Low

The critical requirements change or can not be satisfied, arising the need to rewrite large parts of the project.

Mitigation

Strategy	Avoid
Effectiveness	0.5
Remaining Risk	Very Low

Since we are in control of the requirements for our product we will try not to change them in any breaking way. Additionally, we have already analysed the requirements on a higher level and believe them to be manageable.

Documentation Tools

Damage	Medium
Probability	0.2
Danger	Very Low

The tools used to document the project are inadequate for the documentation of this project.

Mitigation

Strategy	Retain
Effectiveness	0.0
Remaining Risk	Very Low

We have decided to use the recommended documentation tool. We believe it to be adequate. However if it turns out to be inadequate we decided to work around that issue or in other words accept the limitations and work within them.

Development Tools

Damage	Very Low
Probability	0.3
Danger	Very Low

The chosen IDE, VCS etc does not support the development process

Mitigation

Strategy	Reduce
Effectiveness	0.8
Remaining Risk	Very Low

We will set the project up in a way that these factors cannot effect us in any significant way. In fact some developers will be using different IDEs.

Third Party Component

Damage	High
Probability	0.8
Danger	High

A third party component is limited, damaged or otherwise unsuited for our purposes. This includes bugs as well as security vulnerabilities.

Mitigation

Strategy	Reduce
Effectiveness	0.68 ¹
Remaining Risk	Medium

We will evaluate all used third party components before settling with the best match. Additionally, we will abstract the dependency as good as possible with patterns like Repository Pattern and MVVM. This results in significant protection against this risk but cannot mitigate all dangers.

The remaining risk is retained.

Low Software Quality

Damage	Medium
Probability	0.8
Danger	Medium

The quality of the software hinders development, maintainability or adaptability. This will long term reduce the relevance of the project.

¹ An experience value denoting significant chance

The reason why we categorise the damage as “medium” stems from the assumption that this risk would materialise towards the end of the engineering project. At this point we hope to have implemented most important user stories relevant to the project. If however the project would be continued over an extended period of time the damage would could may as well be fatal.

Mitigation

Strategy	Reduce
Effectiveness	0.5
Remaining Risk	Low

Regular refactoring and a strict and rigorous quality control process is hoped to ensure the quality of the software. Besides policies and processes, effective testing should allow us to refactor with more confidence and therefore more often. Furthermore, as mentioned in *Reaching the Complexity Ceiling*, we will invest into the architecture as we believe that good design and the reduction in complexity will be reflected in the overall quality.

Bad User Experience

Damage	High
Probability	0.8
Danger	High

The product does not satisfy the users or customers, resulting in low adoption and - if not mitigated - ends in the death of the project.

This is a rather long term threat to the project but still one that we take very seriously. As we work in a admittedly complicated domain, it is as crucial as it is complicated to achieve good UX.

Mitigation

Strategy	Reduce
Effectiveness	0.68
Remaining Risk	Medium

We have created the role “user experience” and dedicated a member of our team towards the goal of ensuring the usability of our product. We are not confident enough in this mitigation strategy to retain the risk at this point. However, at this point in the project we do not have enough information to decide on further mitigation strategies. Therefore, this risk must be looked out for.

Footnotes

3.3.2 Requirements Engineering

In General we refrain from defining use cases in the requirements engineering in this project. We took a more user centered approach and focus our requirements engineering on epics and user stories. Additionally to the *functional requirements* that are specified in this user centered approach, *non-functional requirements* are also evaluated using the ISO 25010 standard as a reference.

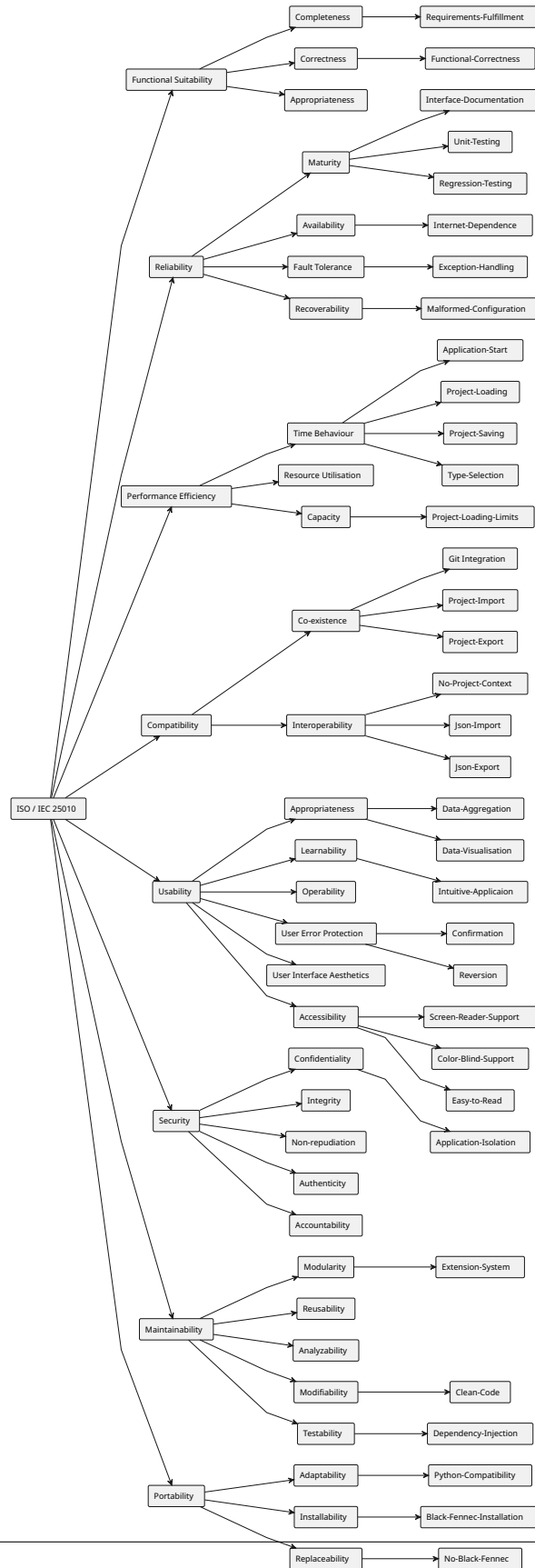
Our user centered approach means, that we created three individual personas that each represent a different user group. These personas can be found [here](#).

Additionally to the traditional requirements, *wireframes of the UI* were created before the start of development to give the viewer an initial understanding of how the application might look like.

For an overview of the contained subsections consult the following table of contents:

Non-Functional Requirements

Overview



Detailed Description

Some points are empty since no non-functional-requirements have been identified yet for these categories. They are marked with the abbreviation TBD, which stands for “To Be Defined”. The titles were left in on purpose to simplify the extensibility of this document.

Functional Suitability

Completeness

TBD.

Correctness

Title (ID)	Functional Correctness
Scenario	A user wants to perform a certain action.
Stimulus	Black Fennec handles the action as expected.
Expectation	Black Fennec provides the correct results with the needed degree of precision.
Measure	Gitlab pipeline does not allow failing test cases. Merge Assignee/Reviewer checks for coverage featured in merge request.
Criteria	At least 90% of the code has to be covered by unit tests which are successful..

Appropriateness

TBD.

Reliability

Maturity

Good Documentation

Title (ID)	Interface Documentation
Scenario	Developer wants to lookup something in the Documentation.
Stimulus	The Developer does not understand how two components interact with each other.
Expectation	Both components and their interaction, if any, can be found in the documentation.
Measure	Merge requests are checked for sufficient documentation of added functionality.
Criteria	Interfaces and classes important to an external developer are documented.

Well Tested

Title (ID)	Unit-Testing
Scenario	A new feature for the application is required.
Stimulus	Developer adds new source code to the project.
Expectation	Code coverage does not decrease significantly. The goal of the new feature is tested on fulfillment. Unit-tests cover equivalence classes and boundaries.
Measure	Merge Request with Assignee and Reviewer prevent code to be merged without Testing.
Criteria	Test coverage above 90 %.

Title (ID)	Regression-Testing
Scenario	A preexistent feature for the application has to be changed.
Stimulus	Developer changes source code of the project.
Expectation	The goal of the feature adaptation is tested on fulfillment. Old usage of the code keeps working as expected. Errors that were fixed before are not reintroduced.
Measure	When bugs are fixed a new testcase has to be created. A CI/CD pipeline test committed code on quality measures and runs the unit-tests of the project.
Criteria	All unit-tests succeed.

Completeness

Title (ID)	Requirements-Fulfillment
Scenario	The Application is used by the client (Team/PO).
Stimulus	A demonstration before the release in the productive environment is done.
Expectation	No unexpected exceptions happen when using the final product. All NFRs required for a successful usage of the application are fulfilled.
Measure	System tests try to measure the fulfillment of functional and non-functional-requirements.
Criteria	Each Requirement if feasible and untested with unit-test is featured in the system-testing protocol to test its fulfillment.

Availability**Internet Dependence**

Title (ID)	Internet-Dependence
Scenario	A user is using the application.
Stimulus	The device used by the user loses connection to the internet.
Expectation	Changes that require to be synchronized to a remote platform can be saved locally and later explicitly uploaded to a shared resource.
Measure	File-sharing is done using git, which comes with capabilities to ensure offline saving and distribution of files in dedicated moments.
Criteria	Any feature requiring internet connection provides offline alternative or meaningful error message.

Fault Tolerance**Handling Exceptions**

Title (ID)	Exception-Handling
Scenario	A user is using the application.
Stimulus	An exception is thrown.
Expectation	The application keeps running. The user is displayed a meaningful error message. The exception is logged, and allow to draw conclusions on why the error happened.
Measure	In code review of merge requests exceptions are looked at to ensure sufficient logging is done.
Criteria	Any exception is captured on application level and printed in a dedicated window.

Recoverability

Title (ID)	Malformed-Configuration
Scenario	A user opens the application.
Stimulus	invalid configuration or invalid module causes a software failure.
Expectation	The application can be started even if the configuration file is malformed.
Measure	A recovery mode (no extensions loaded) allows the loading of valid files which can be parsed, edited and analysed.
Criteria	Application starts despite malformed config.

Performance Efficiency

Time Behaviour

Fast Starter

Title (ID)	Application-Start
Scenario	A user wants to work with the application.
Stimulus	A user opens the application.
Expectation	A loading screen shows the status of the application to the user. As soon as the preparatory tasks are done, the main window opens.
Measure	First operation is showing the loading screen which is capable of showing the status of the operation in progress.
Criteria	The application/loading screen starts within 500ms of clicking the icon on a consumer laptop (intel i5 8th gen + 8gb ram) IF no additional extensions are installed.

Loading Projects

Title (ID)	Project-Loading
Scenario	A user wants to enter data or visualise a file.
Stimulus	A file is loaded by the user.
Expectation	The file opens and the data is displayed in the first meaningful view.
Measure	Only first meaningful view displayed, succeeding operations done afterwards.
Criteria	With a medium sized file (500 MB) it should take no more than 800ms if the reference implementation of the presenter is used.

Quick Save

Title (ID)	Project-Saving
Scenario	A user wants to save the changes made to a file locally.
Stimulus	The user triggers the save option.
Expectation	The changed data is saved into the currently open file.
Measure	Efficient JSON serialisation with dedicated library.
Criteria	Saving a medium sized project with X (TBA) changes takes no longer than 3000ms.

Flash Decision

Title (ID)	Type-Selection
Scenario	A user has data that can be interpreted in multiple ways.
Stimulus	The user selects a type for the visualised data.
Expectation	The visualisation changes to show the selected type.
Measure	Evt. lazy loading to improve performance.
Criteria	With a core data-type it should take no more than 150ms. More advanced types such as lists take no more than 300ms. Extension types are out of scope.

Resource Utilisation

TBD.

Capacity

Heavy Lifter

Title (ID)	Project-Loading-Limits
Scenario	A user wants to enter data or visualise a file of large extent.
Stimulus	A large file is loaded by the user.
Expectation	The file opens and the data is displayed in the first meaningful view.
Measure	Large operations done after showing the first meaningful view. A loading screen also counts as meaningful view.
Criteria	With a large sized file (1 MB) it should be possible to open it in 1000ms if the reference implementation of the presenter is used.

Compatibility

Co-existence

Git Integration

Title (ID)	Git-Integration
Scenario	A user wants share files with other users.
Stimulus	The user prefers the git console over the git-integration in the application and uses it.
Expectation	The application detects changes to its working directory and adjusts relevant data.
Measure	The file system is watched by the application for changes out of scope and copes with them.
Criteria	Changed branches, pulling and conflicts are recognized.
Out of Scope	True

Json as a Service

Title (ID)	Project-Export
Scenario	A user wants to export his project to share with another user including all his settings.
Stimulus	The user exports the project on a certain level (underlay/overlay).
Expectation	The project is exported including all settings of the user.
Measure	Mechanism to create interpretation of project data into exported file. Save Black Fennec version to project file, to allow conversion to newer format.
Criteria	Setting of user compatible with version of importing application are respected.

Title (ID)	Project-Import
Scenario	A user wants to import a project file another user gave him.
Stimulus	The user imports external project.
Expectation	The settings of the project exported are kept in the imported project
Measure	Mechanism to interpret imported project data.
Criteria	Setting of exported project compatible with version of importing application are respected.

Interoperability

Work in External Structure

Title (ID)	No-Project-Context
Scenario	A user wants view/edit file in external directory.
Stimulus	The user opens JSON file with Black Fennec.
Expectation	The file is interpreted and visualised without requiring a project context.
Measure	No dependence on project settings. Check for sufficient rights, understandable error message shown if no permission.
Criteria	Original file is opened if permissions allow. Changes in file can be saved if permissions allow.

Work with files of External Structure

Title (ID)	Json-Import
Scenario	A user wants share files with other users that are encapsulated in directory used by others.
Stimulus	The user imports external json into project.
Expectation	The original file location is saved. The file is copied into the project.
Measure	Check for sufficient rights, understandable error message shown if no permission.
Criteria	Original file is not changed.

Title (ID)	Json-Export
Scenario	A user wants to export a file in his project to an external location.
Stimulus	The user clicks to export a file of the project.
Expectation	The file is exported without containing any Black Fennec proprietary data.
Measure	Check for sufficient rights, understandable error message shown if no permission.
Criteria	File at location is overwritten. No proprietary data contained in exported file.

Usability

Appropriateness

Data Aggregation

Title (ID)	Data-Aggregation
Scenario	A user wants to aggregate data from several sources into one file.
Stimulus	The user open a new project and inputs external data.
Expectation	The application allows the user an efficient workflow for aggregation of data.
Measure	Dedicated presenters for different use cases to allow optimized workflows.
Criteria	The most important functions are maximum two clicks away.

Data Visualisation

Title (ID)	Data-Visualisation
Scenario	A user wants to look at interconnected data.
Stimulus	The user opens a file containing interconnected data.
Expectation	The application shows an overviewable visualisation of interconnected data.
Measure	Dedicated presenter for visualisation of interconnected data (graph).
Criteria	interconnection of data visualised with lines in between data.
Out of Scope	True

Learnability

Just Like an Apple

Title (ID)	Intuitive-Application
Scenario	A user wants to work with Black Fennec.
Stimulus	The user opens the application for the first time.
Expectation	The user can operate basic use cases after few minutes of using the application.
Measure	Walk-through upon first opening of application. Manual for usage of application.
Criteria	Closed-card-sort and tree-sort passed with industry standards.

Operability

TBD.

User Error Protection

Better than Hawaii

Title (ID)	Confirmation
Scenario	A user wants to perform a critical operation.
Stimulus	The user clicks to perform a critical operation.
Expectation	The user is asked whether he is not accidentally click said action.
Measure	Show confirmation dialog before executing critical operation.
Criteria	Confirmation dialog before performing critical actions.

Title (ID)	Reversion
Scenario	A user wants to perform a critical operation.
Stimulus	The user clicks to perform a critical operation.
Expectation	The user is able to undo the critical action for a specified amount of time.
Measure	Save previous state of application to rollback.
Criteria	Critical actions can be roll-backed for at least 1m if possible.

User Interface Aesthetics

TBD.

Accessibility

Stolze Spezial

Title (ID)	Screen-Reader-Support
Scenario	A user wants to understand the interface without seeing it.
Stimulus	A user triggers the screen reading function.
Expectation	The screen reader understands the software interface and can translate text into speech.
Measure	All main functionalities are equipped with the text to speech "tag".
Criteria	System Test with impaired person succeeds.
Out of Scope	True

Title (ID)	Color-Blind-Support
Scenario	A user wants to understand any clickable action despite his color-blindness.
Stimulus	A color blind user uses Black Fennec.
Expectation	Clickable actions are distinguishable besides their color. 2-Senses principle.
Measure	2-Senses principle adhered in UI design.
Criteria	Black-and-White Test of the application.
Out of Scope	True

Title (ID)	Easy-to-Read
Scenario	A user wants to understand the interface despite minor visual impairment
Stimulus	A user with minor visual impairment uses Black Fenec.
Expectation	If the font size is too small to see, it can be resized. Contrast of colors make it easy to read text.
Measure	AA-Rating in color contrast. Resizable font size, and responsive design to cope with big text.
Criteria	Font size customizable. Color Palette checked for AA-Rating.
Out of Scope	True

Security

The security section is not filled with many NFRs because the assessment of these NFRs is done via Threat-analysis and Attack-Trees.

Confidentiality

Sand Box

Title (ID)	Application-Isolation
Scenario	A user imports malicious data into Black Fenec.
Stimulus	The Malicious code is executed inside the Black Fenec tool.
Expectation	The malicious data doesn't affect the OS.
Measure	Application can be executed in isolated environment.
Criteria	Application is sand boxed.
Out of Scope	True

Integrity

TBD.

Non-repudiation

TBD.

Authenticity

TBD.

Accountability

TBD.

Maintainability**Modularity**

Title (ID)	Extension-System
Scenario	A Developer wants to develop additional feature.
Stimulus	Additional feature required.
Expectation	Easily usable interface to extend functionality of Black Fenec.
Measure	Extension Manager providing extension API which allow for extensions.
Criteria	Possibility to extend functionality exists.

Reusability

TBD.

Analyzability

TBD.

Modifiability**Windows is Broken. Long live Windows**

Title (ID)	Clean Code
Scenario	A Developer develops low quality code for the sake of time.
Stimulus	The general code quality decreases.
Expectation	Developers pay attention to clean code and broken widows in development
Measure	Code will be reviewed before every merge.
Criteria	Can be measured using pylint.

Testability

Title (ID)	Dependency-Injection
Scenario	Developer has to test component to achieve sufficient coverage.
Stimulus	Developer tests code.
Expectation	The code written by the developer allows for easy testing by mocking component.
Measure	Components are linked through dependency injection.
Criteria	90% test coverage is maintained.

Portability

Adaptability

Title (ID)	Python-Compatibility
Scenario	A user wants to install Black Fennec
Stimulus	User executes Black Fennec
Expectation	The program runs independent of the OS. Any system with python installed can run Black Fennec
Measure	Application written in Python
Criteria	Application runs on python

Installability

Title (ID)	Black Fennec-Installation
Scenario	A user wants to install Black Fennec via the command line.
Stimulus	The user executes the pip install. . . command.
Expectation	The user can install the tool using the pip install command.
Measure	The user can start the Black Fennec Tool via the desktop icon.
Criteria	Application installable via pip CLI.

Replaceability

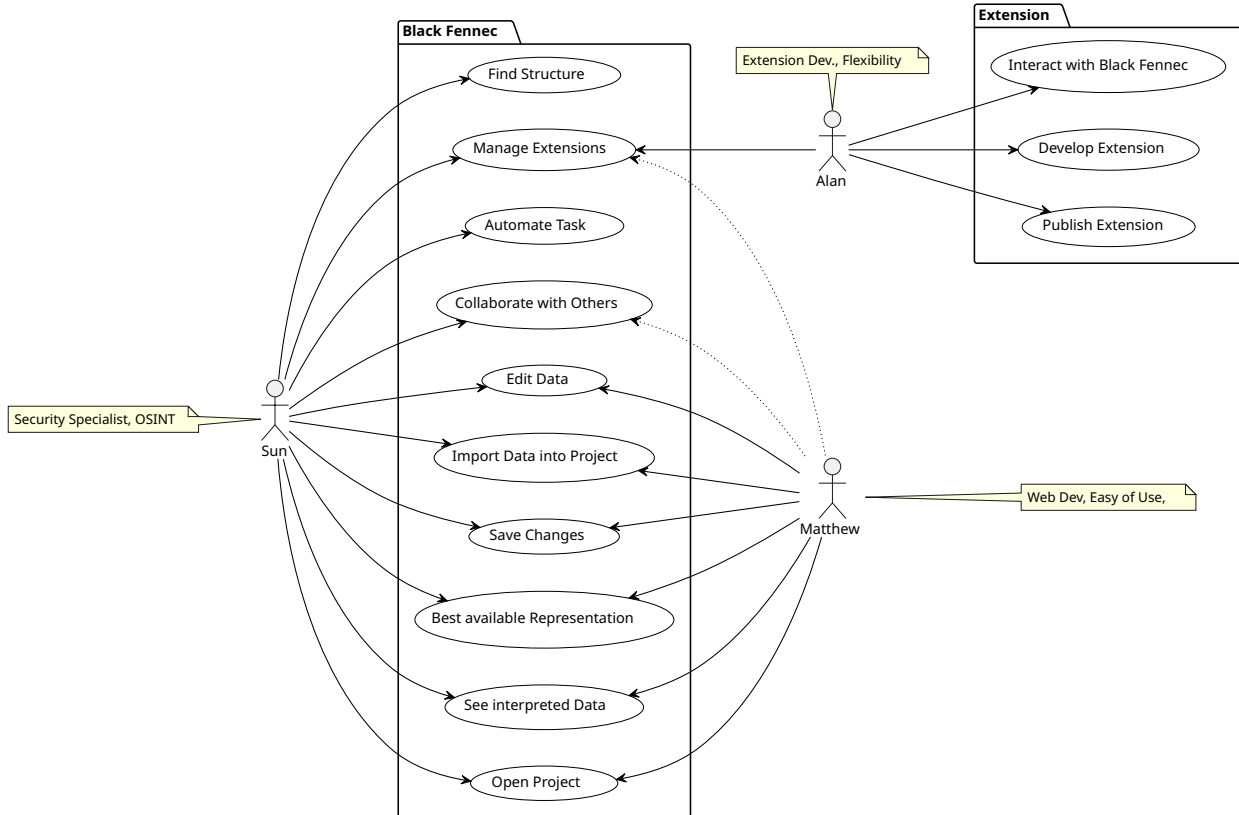
Hail JSON

Title (ID)	No-Black-Fennec
Scenario	The user wants to read files created with Black Fennec without Black Fennec.
Stimulus	Black Fennec file opened with plain text editor.
Expectation	The file produced by Black Fennec is readable to the user. Saved data unpolluted with internal states, overviewability of JSON kept.
Measure	All data is saved as JSON and therefore is easily readable.
Criteria	Files created with Black Fennec are JSON.

Functional Requirements

Epic Overview

This diagram provides an overview over the epics that our specified *Personas* would like to be able to do with Black Fennec. It is intentionally done like an UML diagram overview but is apart from the format quite different. The personas are not representing specific roles that have to be able to do something but a category of users that have certain needs and expectations.



User Stories

The user epics described in this document can be found in the github repository in the [issue board](#) and can be distinguished by their unique formulation which is structured like follows: *As <persona> i want ... so that*

These epics can then be broken down into more detailed parts called user stories if necessary, which are created in the range of sprint plannings or a dedicated backlog-refinement meeting.

Personas

Matthew G. Gritton



Age: 52

Profession: self taught web developer (freelancer)

Usertyp: Late Majority

Defining Traits: superficial knowledge, impatient

Wants & Needs:

- Provide high quality web solutions for small businesses
- Working with extendable tools instead of many specific
- Using tools where no deeper knowledge is needed

Frustrations:

- New Tools are often hard to understand for him, especially if they lack in UI/UX design

Sun Chuang



Age: 26

Profession: Security specialist

Usertyp: Early Adopter

Defining Traits: Self-Motivated, perfectionist, curious

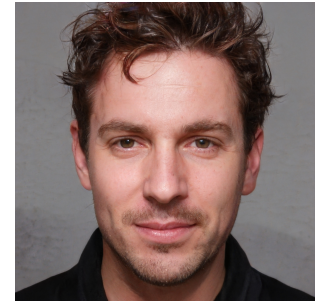
Wants & Needs:

- Information gathering for her research
- automate tasks and analyze data

Frustrations:

- Commonly used OSINT tools don't fit her needs
- Additional work needed to compensate for missing features of available tools

Alan Nordin



Age: 34

Profession: Extension developer

Usertyp: Early Majority

Defining Traits: Team player, business-oriented

Wants & Needs:

- Develop extensions for universal use
- publish extensions easily and receive feedback

Frustrations:

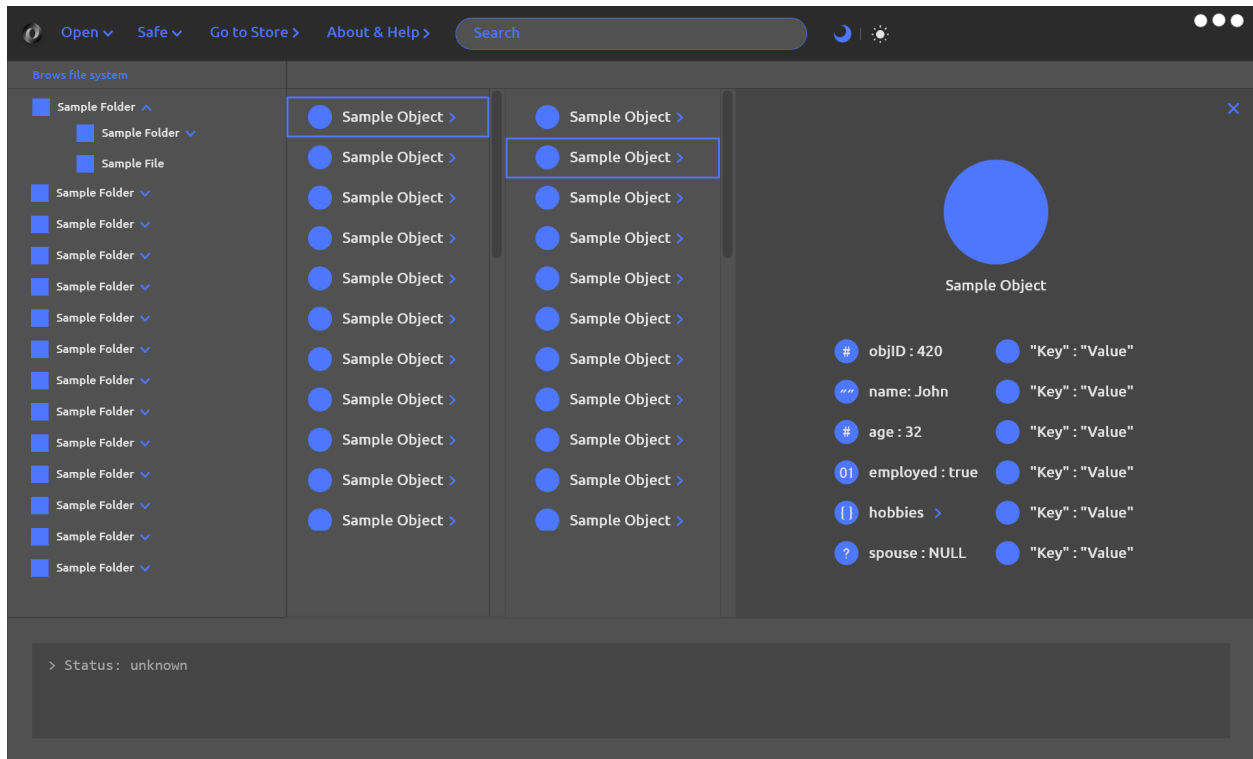
- Many available tools are unflexible and hard to extend

UI Sketches

Black Fennec GUI

The Graphical User Interface (GUI) of Black Fennec is very simple and user-friendly. There is a menu in the upper area. There the user can open, create or implement projects. The user can save projects. The extension marketplace can be opened via the “go to store” button. Opened projects can be searched by using the search engine. As a small feature, we are implementing a selection between default and dark mode.

Under the menu bar the user can see an overview of the imported data.



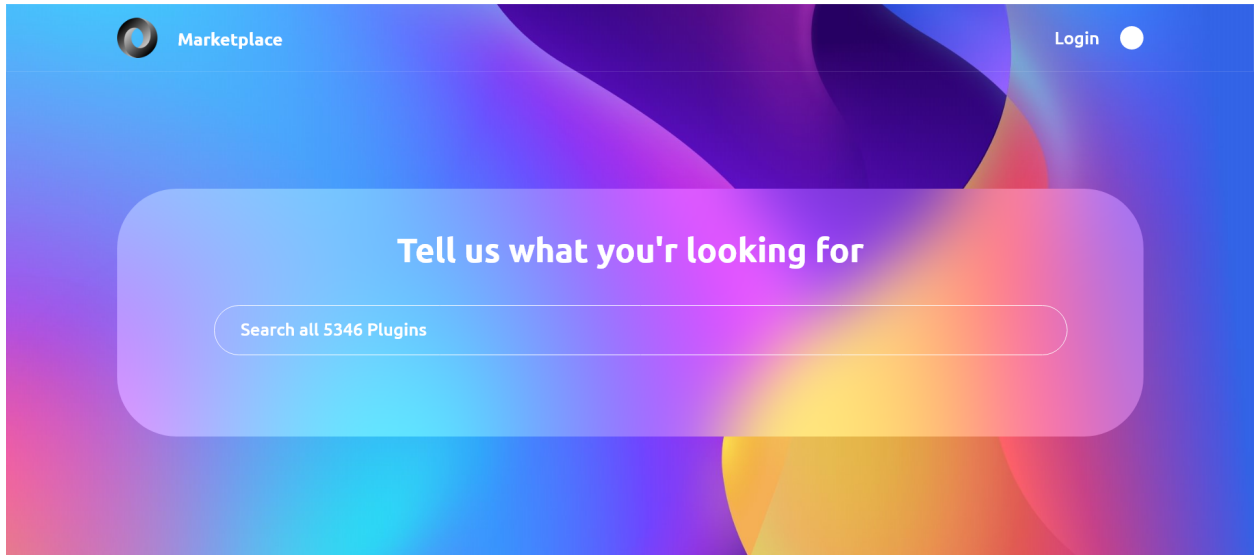
Black Fennec Main UI and structure presenter: Object detail view containing sample key-value pairs

Extension Marketplace

During the creation of the UI sketches we thought that we could create an external website for the extension marketplace, which we then decided against. In place of the external extension store up to version 0.6 an internal extension store was implemented. This was then rendered obsolete by the use of flatpak extensions from version 0.11 onwards. Thanks to the flatpak extension system, Black Fennec no longer needs an internal extension store.

For the sake of completeness - and maybe future reference - the envisioned marketplace is still included.

Initially the website shows an overview of all available extensions and includes a searchengine.



Staff Picks

Three plugin cards are displayed under the "Staff Picks" section. Each card consists of a circular profile picture placeholder, the text "Sample Plugin", a row of five blue dots (the first four are filled, the fifth is empty), a short paragraph of Lorem Ipsum text, and a "69'420 downloads" label. At the bottom right of each card is a blue "Download" button.

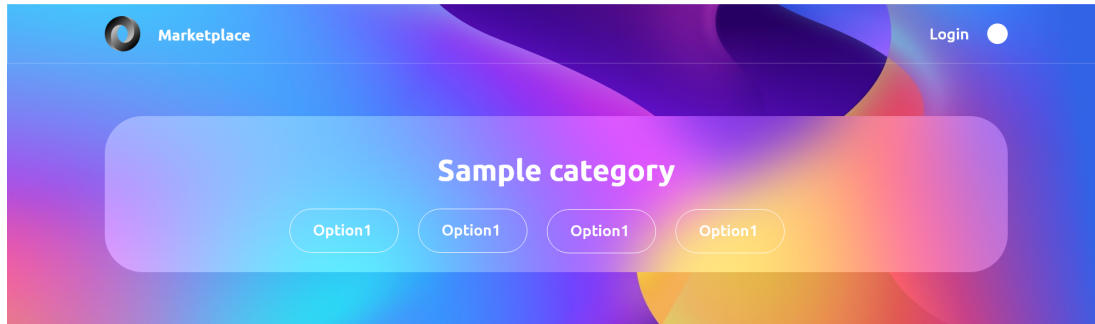
Top Downloads

Three plugin cards are displayed under the "Top Downloads" section. Each card is identical in layout to the ones in the "Staff Picks" section, featuring a circular profile picture placeholder, the text "Sample Plugin", a row of five blue dots (the first four are filled, the fifth is empty), a short paragraph of Lorem Ipsum text, and a "69'420 downloads" label. At the bottom right of each card is a blue "Download" button.

Brows by category

A horizontal row of five rounded rectangular buttons with blue borders and white backgrounds. The first button contains the text "All categories". The next three buttons each contain the text "Sample category". Below this row, there is a single "Sample category" button. To the right of these buttons is a blue circular button with a white upward-pointing arrow.

Black Fennec Extension Store: Main UI



Staff Picks

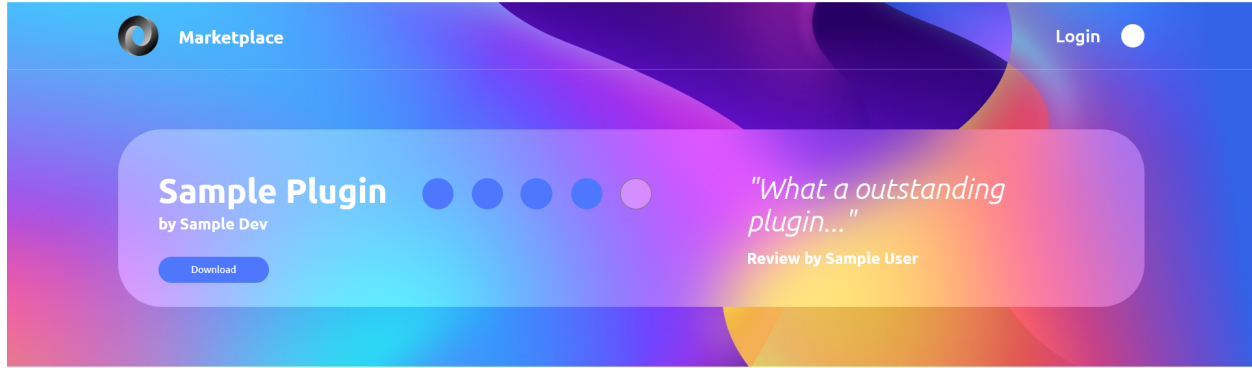
<p>Sample Plugin</p> <p>69420 downloads</p> Download	<p>Sample Plugin</p> <p>69420 downloads</p> Download	<p>Sample Plugin</p> <p>69420 downloads</p> Download
<p>Sample Plugin</p> <p>69420 downloads</p> Download	<p>Sample Plugin</p> <p>69420 downloads</p> Download	<p>Sample Plugin</p> <p>69420 downloads</p> Download
<p>Sample Plugin</p> <p>69420 downloads</p> Download	<p>Sample Plugin</p> <p>69420 downloads</p> Download	<p>Sample Plugin</p> <p>69420 downloads</p> Download
<p>Sample Plugin</p> <p>69420 downloads</p> Download	<p>Sample Plugin</p> <p>69420 downloads</p> Download	<p>Sample Plugin</p> <p>69420 downloads</p> Download

Show More

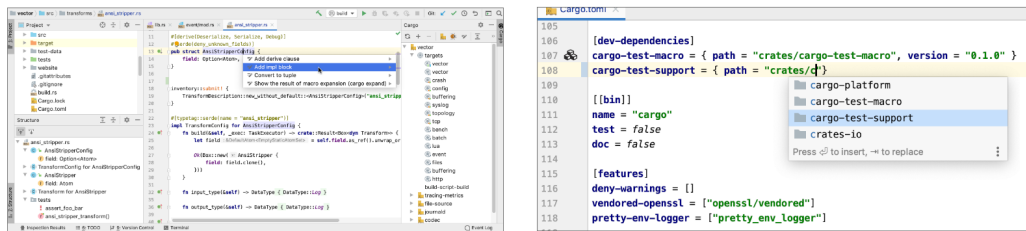


Black Fennec Extension Store: Browse by category sample page

By selecting the desired extension, the user can see an overview of the extension and read a small description of it.



Overview Versions Reviews



Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.

What's New

1.4.30

Preview of new language features: JVM records support, sealed interfaces, and stable inline classes.

Kotlin/JVM: IR backend is now in Beta.

Kotlin/Native: performance improvements, new `watchosX64` simulator target, support for Xcode 12.2 libraries.

Kotlin/JS: prototype lazy initialization of top-level properties.

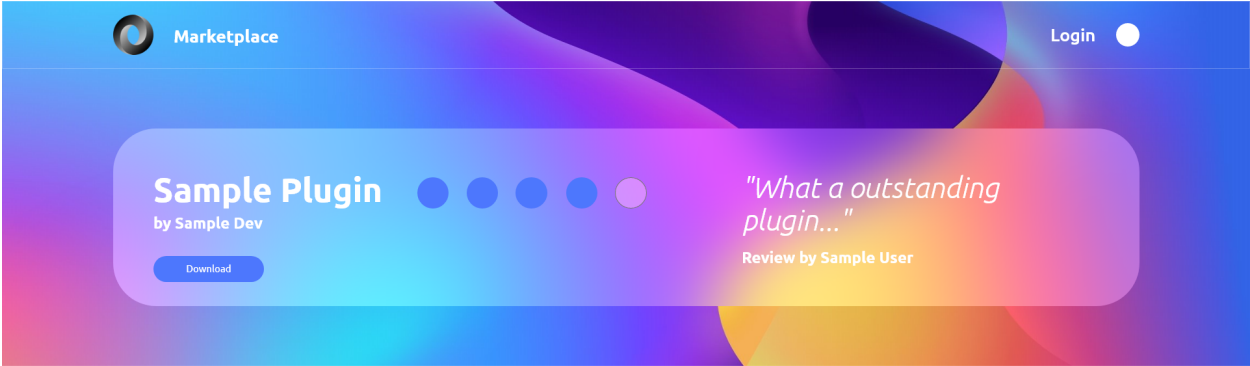
Support for Gradle configuration cache.

Standard library API improvements: locale-agnostic API for upper/lowercasing text and clear Char-to-code and Char-to-digit conversions.

For more details, see What's New in Kotlin 1.4.30 and this blog post.



Black Fennec Extension Store: Sample plugin overview



Overview **Versions** Reviews

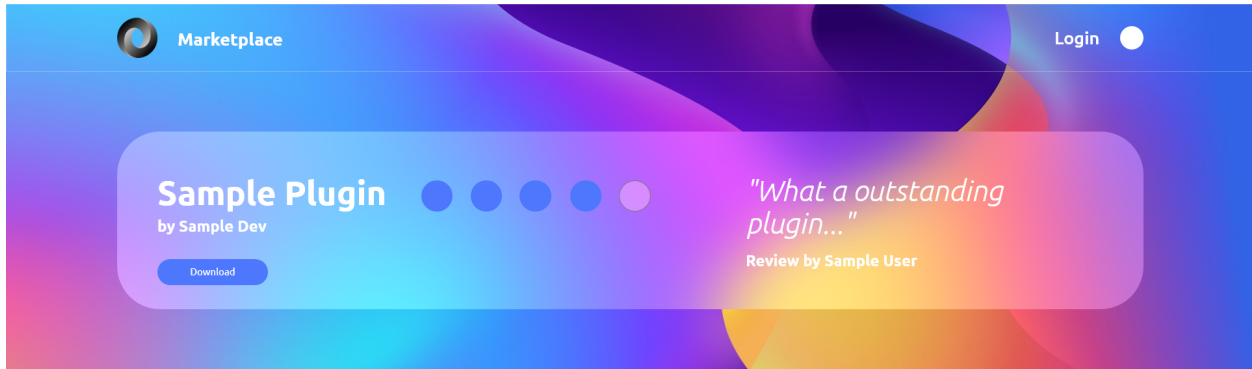
Version History

Version	Update Date	
202-1.4.31-release-AS8194.7	25. Feb. 2021	Download
202-1.4.31-release-AS8194.7	25. Feb. 2021	Download
202-1.4.31-release-AS8194.7	25. Feb. 2021	Download
202-1.4.31-release-AS8194.7	25. Feb. 2021	Download
202-1.4.31-release-AS8194.7	25. Feb. 2021	Download
202-1.4.31-release-AS8194.7	25. Feb. 2021	Download
202-1.4.31-release-AS8194.7	25. Feb. 2021	Download
202-1.4.31-release-AS8194.7	25. Feb. 2021	Download

Show More



Black Fennec Extension Store: Sample plugin Version History



Overview Version **Reviews**

Ratings & Reviews

4.0 out of 5

Sample User 14. Feb. 2021

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Sample User 14. Feb. 2021

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Sample User 14. Feb. 2021

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Show More



Black Fennec Extension Store: Sample plugin ratings and reviews

3.3.3 Project Standards

This document gives an overview of the standards that developers shall try to adhere to when contributing to the project. Also it lists the quality control measures taken in this project. We will reference dedicated documents which are considered single-source-of-truth on their respective topics and we recommend reading them for a better grasp of all active policies.

Quality has been one of our top concerns from the get-go and we have planned for it accordingly. We deploy multiple techniques to ensure the quality of our product - including its documentation.

One of the first artifacts produced was our *quality assurance*, in which we outlay in detail the measures we take to ensure the high quality we require.

For an overview of the standards we follow, refer to the following documents:

Quality Assurance

To ensure the desired quality in this project many different standards are enforced.

Produced Artifacts

All created artifacts of this project are contained within the project Gitlab repository to which any associated person has access. This ensures that of every file a detailed change history is available. For a detailed elaboration on our Version Control Strategy a dedicated document *version control strategy* exists.

Documentation

Our documentation is written in the Markdown-flavour RestructuredText and is version controlled in Gitlab. The generation and provision of the Documentation is automated through a CI-Pipeline task and is automatically built when attaching a tag to a commit. The documentation also is a project artifact and therefore also falls under the *version control strategy*.

Documentation Guidelines

Changes to the documentation are handled like any other changes to any other artifact with the addition of work items specifically to ensure consistency and relevance. We utilise our usual review processes to ensure correctness and the communication of knowledge. This entails the inclusion of the work done in sprint reviews and retrospectives as outlined in our *definition of scrum*.

The Language used in the documentation of this project is english. Created it is with the Sphinx framework and RestructuredText. Before a commit the developer has to build the document on his local system to ensure the build passes successfully. For this a makefile is provided which can be executed with `make docs`.

The documentation is written with the form follows function principle in mind. The software as a product is the main goal of this project and a functional documentation can be written more efficiently.

Project Management

Project Management is done with Gitlab as well. The Issue-Boards are used in this project to represent our different steps an issue can undergo. The ordering in the lists show the importance of each issue. The higher up an issue is, the higher is its importance. The issues created can be assigned to members of the team and to the milestone that they belong to. To know to which sprint an issue is associated with, dedicated Sprint labels are created at the start of a sprint.

The different steps an issue can undergo are described in the following table.

Column (Label)	Description
Roles	These issues represent different roles in which administrative effort can be captured.
Draft	The drafted issues are in a pre-stage before making it into the backlog. Here any member can enter ideas that come up during sprints and are then evaluated by the Product Owner
Backlog	Only the Product Owner is allowed to define the Backlog. Here Definite Tasks that are going to be implemented are listed.
Sprint Backlog	This Column is the Backlog for the Sprint and requires the items to be actual user-stories and on a Work-item level so that one person can be assigned to this task.
In Progress	In this list are issues that are in progress in the current sprint.
Resolved	When an issue is resolved, this means that it adheres to the <i>definition of done</i> but was not merged yet.
Open	The Open Issue list is not used in our project as open issues tend to reside in dedicated lists.
Closed	When an feature was successfully merged its issues move to the list closed where all past issues reside.

Development

Procedure

As previously mentioned all our artifacts including our code is contained in Gitlab. For any versioning specific practices one can look into the *version control strategy*. In there the handling of merge requests is explained. This ensures a dual control principle regarding not only logic errors but also formatting and code style.

Additionally to the dual control principle our Gitlab pipelines have built in execution for testing and linting.

Code Style Guidelines

To support well-formatted coding, every team member is required to install `pylint`, a linting tool that not only can enforce Coding Standards according to Python's PEP8 style guide but also features an error detection and refactoring help. Additionally a pipeline task that includes `pylint` checking is included in the ci on Gitlab. A Script is used to convert the `pylint` error codes into a rating, preventing the task to fail from notices other than errors. The `.pylintrc` file is copied from the [Google Styleguide](#). But it is intended to update the `pylintrc` configuration file to exclude warning generating rules, if the whole team approves.

Code Reviews

Our workflow is heavily inspired by the gitflow work-flow as documented in our *version control strategy* and includes *merge requests*. Each merge request is reviewed by at least one other developer to not only ensure quality but also spread domain knowledge as evenly as possible. The assignee and reviewer additionally should read the code and try to understand it, and if unclear ask the creator.

Additionally, we deploy automated reviews and enforce *code style guidelines* `<code_style_guidelines>` with the help of `pylint`.

Pair Programming

An important tool in our project will be the use of pair programming. It will not be done for every function that is written but if anyone is stuck and is only progressing slowly this shall be recognized during our daily Scrum meetings and another member of the team will try to help resolve the problem with a fresh mind.

Testing

Stability and regression safety is at the forefront of quality control. This is precisely why we uphold very high testing standards in this project. We encourage our developers to take the time to achieve 100% coverage for non-UI code. To ensure our high standards we configured our CI/CD pipeline to strictly require a branch coverage of 90% or more as well as the passing of all available tests. As a result our test coverage is constantly around 94%.

If you are interested in the documentation of coverage and failed pipelines, we suggest visiting the gitlab built-in analytics for [pipelines](#) and [coverage](#).

The tests are stored in a separate location in the tests folder. There each python file of our application that is tested has a corresponding testing file that preferably has the same name if no ambiguities are present.

A doubles folder exists where commonly used doubles are saved in specific files separated by component they belong to.

Unit Testing

Unit tests are done with the `pytest` framework in our project. We value test-driven-development and strive for 90% of testing coverage. Every developer is ought to have `pytest` installed in his development environment and additionally a pipeline ensures that all tests are always passing. A merge request without passing pipelines is prohibited from being merged.

Integration Testing

Fewer in number but equally important are our integrations tests. These tests ensure that the components tested with unit tests also work together.

Integration Tests are done in the PyTest framework as well. They reside in separate files with an "_integration" addition to their filename and should be marked as integration tests with the pytest decorator `@pytest.mark.integration`. Since the integration tests are mostly contained in separate files, it is also possible to mark the whole file with `pytestmark = pytest.mark.integration`.

System Tests

To ensure the integrity of the system as a whole, we conduct system tests before releasing a new version. We restrain from executing the system test suit for every merge request, because they are very time consuming and cover most aspects of the applications normal execution, most of which are likely to remain unchanged. As a replacement we rely on manual, undocumented tests done by the developer and reviewer.

Since the System Tests are not automated, they are written down in a specific test protocol that is created using the *system test template*. This way the tests can be done multiple times using the same template. As with the documentation of the project it is expected to append the test protocol when adding new functionality.

The protocols are contained in a single file that can be found [here](#). This file includes all recent releases and their respective test protocols.

Usability Tests

On top of all the functional testing conducted as described above, we also undertake usability studies. These tests are usually performed at the end of a project with the goal of having each *persona* represented by at least one participant. The templates can found [here](#) and the protocols [here](#).

Definition of Done

For each work item an issue **MUST** be created. An issue is considered done IFF it's corresponding commits have been successfully merged into the dev branch. Furthermore, the time spent **MUST** be documented.

Merging into dev Branch

A merge request **MUST** be created for merging into the dev branch.

Before any such request can be accepted, merge requests **MUST**:

- passing of pipeline
- test coverage is above 90 percent
- be reviewed by another developer
- have all conversations resolved

Merge requests **SHOULD**:

- NOT lower test coverage
- confirm to the projects *Logging Standards*

Definition of Scrum

This document defines the definition of Scrum used in this project. This document can be edited during e.g. a *sprint retrospective*

Daily Scrum

This meeting usually takes place once a day will should last about 15 minutes. We collectively decided to hold an additional meeting of this type. It is mainly intended for the developers of our team. During the meeting the progress towards the Sprint Goal will be reviewed. If necessary, the Sprint Backlog and the planned work can be adjusted.

Sprint Planning

This meeting is used to plan a new sprint and takes place as soon as a Sprint has been completed.

Three main Questions are discussed in this meeting. These include defining a new Sprint Goal, which items from the Product Backlog should be included in the upcoming Sprint and how an increment that meets the Definition of Done can be created.

Sprint Review

During this meeting, the result of the Sprint is reviewed and adjustments are made.

Our team presents their work and progress towards the Product Goal. Based on the information, the next steps can be planned.

Sprint Retrospective

This meeting is very important. It is used to plan how to improve the quality and effectiveness of our work. The team reviews how the last sprint went in terms of individual team members, interactions, processes, tools and their Definition of Done. We discuss, what went well during the Sprint and what didn't, what problems were encountered and how they were solved or can be solved.

Milestone Meeting

After each milestone there is a meeting with the Advisor. We present a small demo of our project and show the status of the product.

Logging Standards

This document describes the logging practices currently active in this project. The described practices and standards do not apply to code written when other logging standards were active.

Logging Practice

- Exceptions, both handled and re-thrown, MUST be logged.

Version Control Strategy

This project does version control using Gitlab according to the git Workflow. Before a feature branch can be merged a merge request MUST be created.

Merge Requests

There MUST always be two members of the project assigned to a merge request. An assignee who has the competence to merge and a reviewer who both are obligated to check whether the commits fulfill the *Definition of Done*. The Assignee is assigned to the “quality assurance” role, and the reviewer depends on the topic the merge request is about. In this project code mainly regarding UI is assigned to the “user experience” role, any other code to the “architecture” role. Changes solely regarding the documentation are assigned to the “documentation” role.

Approval

Both assigned roles SHOULD give their approval to a merge request. The assignee however can merge without the approval of the reviewer IFF the reviewer exceeds the maximal time of one week without giving his approval and without starting a conversation.

Comments

Assigned roles can, if the quality does not suffice open a communication by writing a comment. These comment must be resolved before the merge request can be merged.

Coverage

The coverage in a merge request can be seen in Gitlab. A pipeline automatically uploads a coverage report created with the tool pytest. The coverage of a merge request SHOULD not lower the previously established coverage, but in some cases this cannot be hindered. But in any case the coverage SHALL never be lower than 90 Percent.

Pipelines

The Gitlab pipelines of this project are to be split into separated folders to improve the overviewability, and pipeline run-times when making changes to files significant to the rebuilding of the building docker images.

Building docker images

To improve pipeline run times a docker in docker build is done for the generation of the documentation and the testing and deployment of the project. These build images that have installed the required packages to be able to fulfill their respecting task are uploaded into the Gitlab container registry from where they are pulled for succeeding tasks.

Deployment

The deployment of the project and associated extensions are uploaded into the PyPI package registry also in the Gitlab repository. From there they can be downloaded using the Python package management tool pip.

Design Standards

The Project tries to accommodate the [GNOME Human Interface Guidelines](#) as much as possible. If any deviations occur from the HIG, feel free to open an issue or pull request to discuss it.

3.4 Index

PYTHON MODULE INDEX

b

blackfennec, 149
blackfennec.action_system, 105
blackfennec.action_system.action, 104
blackfennec.action_system.action_registry, 104
blackfennec.action_system.context, 104
blackfennec.document_system, 114
blackfennec.document_system.document, 113
blackfennec.document_system.document_factory, 114
blackfennec.document_system.document_registry, 114
blackfennec.document_system.mime_type, 110
blackfennec.document_system.mime_type.in_memory, 105
blackfennec.document_system.mime_type.in_memory.in_memory_mime_type, 105
blackfennec.document_system.mime_type.json, 108
blackfennec.document_system.mime_type.json.json_mime_type, 105
blackfennec.document_system.mime_type.json.json_pointer_serializer, 106
blackfennec.document_system.mime_type.json.json_reference_serializer, 107
blackfennec.document_system.mime_type.mime_type, 108
blackfennec.document_system.mime_type.mime_type_registry, 109
blackfennec.document_system.resource_type, 113
blackfennec.document_system.resource_type.protocols, 112
blackfennec.document_system.resource_type.protocols.binary_resource_type, 110
blackfennec.document_system.resource_type.protocols.file_resource_type, 110
blackfennec.document_system.resource_type.protocols.https_resource_type, 111
blackfennec.document_system.resource_type.resource_type, 112
blackfennec.document_system.resource_type.resource_type_registry, 112
blackfennec.extension_system, 116
blackfennec.extension_system.extension, 114
blackfennec.extension_system.extension_api, 115
blackfennec.extension_system.extension_registry, 116
blackfennec.extension_system.extension_service, 116
blackfennec.layers, 123
blackfennec.layers.encapsulation_base, 119
blackfennec.layers.encapsulation_base.base_factory_visitor, 116
blackfennec.layers.encapsulation_base.encapsulation_base, 117
blackfennec.layers.encapsulation_base.list_encapsulation_base, 118
blackfennec.layers.encapsulation_base.map_encapsulation_base, 118
blackfennec.layers.encapsulation_base.reference_encapsulation_base, 119
blackfennec.layers.merge, 122
blackfennec.layers.merge.merged_layer, 119
blackfennec.layers.merge.merged_list, 120
blackfennec.layers.merge.merged_map, 120
blackfennec.layers.merge.merged_null, 120
blackfennec.layers.merge.merged_phantom, 120
blackfennec.layers.merge.merged_structure, 120
blackfennec.layers.merge.merger, 121
blackfennec.layers.observable, 123
blackfennec.layers.observable.observable, 122
blackfennec.layers.observable.observable_base, 122
blackfennec.layers.observable.observable_factory_visitor, 123
blackfennec.layers.overlay, 123
blackfennec.layers.overlay.overlay, 123
blackfennec.layers.overlay.overlay_base, 123
blackfennec.layers.overlay.overlay_factory_visitor, 123

blackfennec.presentation_system, 131
 blackfennec.presentation_system.about_window, 124
 blackfennec.presentation_system.about_window.about_window_view, 124
 blackfennec.presentation_system.about_window.about_window_view_model, 124
 blackfennec.presentation_system.extension_warning_dialog, 124
 blackfennec.presentation_system.extension_warning_dialog.extension_warning_dialog, 124
 blackfennec.presentation_system.history_service, 128
 blackfennec.presentation_system.main_window, 126
 blackfennec.presentation_system.main_window.black_fennec_view, 124
 blackfennec.presentation_system.main_window.black_fennec_view_model, 124
 blackfennec.presentation_system.main_window.document_tabs, 125
 blackfennec.presentation_system.main_window.document_tabs_view, 126
 blackfennec.presentation_system.main_window.file_columns_view, 126
 blackfennec.presentation_system.navigation_service, 127
 blackfennec.presentation_system.navigation_service.navigation_service, 127
 blackfennec.presentation_system.presenter_registry, 129
 blackfennec.presentation_system.structure_view_factory, 129
 blackfennec.presentation_system.type_view, 130
 blackfennec.presentation_system.type_view_factory, 130
 blackfennec.presentation_system.type_view_factory.black_fennec_registry, 130
 blackfennec.presentation_system.ui_service, 128
 blackfennec.presentation_system.ui_service.message, 128
 blackfennec.presentation_system.ui_service.minimal_type_select_type_system, 127
 blackfennec.presentation_system.ui_service.minimal_type_select_type_system.type_coverage_mixin, 127
 blackfennec.presentation_system.ui_service.minimal_type_select_type_system.type_reflection_view, 127
 blackfennec.presentation_system.ui_service.ui_service, 128
 blackfennec.structure, 137
 blackfennec.structure.boolean, 133
 blackfennec.structure.list, 133
 blackfennec.structure.map, 134
 blackfennec.structure.null, 135
 blackfennec.structure.number, 135
 blackfennec.structure.reference, 135
 blackfennec.structure.reference_navigation, 136
 blackfennec.structure.reference_navigation.child_navigator, 136
 blackfennec.structure.reference_navigation.index_of_navigator, 136
 blackfennec.structure.reference_navigation.navigator, 136
 blackfennec.structure.reference_navigation.parent_navigator, 136
 blackfennec.structure.reference_navigation.root_navigator, 136
 blackfennec.structure.reference_navigation.sibling_offset, 136
 blackfennec.structure.reference_navigation.uri_navigator, 136
 blackfennec.structure.string, 136
 blackfennec.structure.structure, 136
 blackfennec.structure.structure_serializer, 136
 blackfennec.structure.visitor, 137
 blackfennec.type_system, 146
 blackfennec.type_system.boolean_type, 141
 blackfennec.type_system.coverage, 141
 blackfennec.type_system.interpretation, 141
 blackfennec.type_system.interpretation.coverage, 141
 blackfennec.type_system.interpretation.interpretation, 141
 blackfennec.type_system.interpretation.interpretation_service, 141
 blackfennec.type_system.interpretation.offer, 141
 blackfennec.type_system.interpretation.specification, 141
 blackfennec.type_system.list_type, 141
 blackfennec.type_system.map_type, 142
 blackfennec.type_system.null_type, 142
 blackfennec.type_system.number_type, 142
 blackfennec.type_system.reference_type, 143
 blackfennec.type_system.string_type, 143
 blackfennec.type_system.type, 143
 blackfennec.type_system.type_coverage_mixin, 144
 blackfennec.type_system.type_loader, 145
 blackfennec.type_system.type_parser, 145
 blackfennec.type_system.type_registry, 145
 blackfennec.util, 149
 blackfennec.util.change_notification, 146

blackfennec.util.change_notification_dispatch_mixin, 146

blackfennec.util.comparable, 146

blackfennec.util.deep_copy, 146

blackfennec.util.intercepting_visitor, 147

blackfennec.util.meta_info, 147

blackfennec.util.observable, 148

blackfennec.util.parameterized_visitor, 148

blackfennec.util.service_locator, 148

blackfennec.util.type_comparator, 148

blackfennec_doubles, 170

blackfennec_doubles.action_system, 150

blackfennec_doubles.action_system.double_action_registry, 149

blackfennec_doubles.action_system.double_action_registry, 149

blackfennec_doubles.action_system.double_context, 149

blackfennec_doubles.action_system.double_ui_controller, 150

blackfennec_doubles.document_system, 154

blackfennec_doubles.document_system.double_document, 154

blackfennec_doubles.document_system.double_document_factory, 154

blackfennec_doubles.document_system.double_document_registry, 154

blackfennec_doubles.document_system.mime_type, 153

blackfennec_doubles.document_system.mime_type.double_mime_type, 152

blackfennec_doubles.document_system.mime_type.double_mime_type_registry, 152

blackfennec_doubles.document_system.mime_type.json, 152

blackfennec_doubles.document_system.mime_type.json.double_json_pointer_parser, 150

blackfennec_doubles.document_system.mime_type.json.double_json_reference_serializer, 151

blackfennec_doubles.document_system.mime_type.json.double_structure_serializer, 151

blackfennec_doubles.document_system.resource_type, 154

blackfennec_doubles.document_system.resource_type.double_resource_type, 153

blackfennec_doubles.document_system.resource_type.double_resource_type_registry, 153

blackfennec_doubles.double_dummy, 170

blackfennec_doubles.extension_system, 157

blackfennec_doubles.extension_system.double_extension, 156

blackfennec_doubles.extension_system.double_extension_loader, 156

blackfennec_doubles.extension_system.double_extension_registry, 157

blackfennec_doubles.extension_system.double_extension_source, 157

blackfennec_doubles.extension_system.double_extension_source, 157

blackfennec_doubles.extension_system.double_extensions, 156

blackfennec_doubles.extension_system.double_extensions_creator, 155

blackfennec_doubles.extension_system.double_extensions_creator, 155

blackfennec_doubles.extension_system.double_extensions_descriptor, 155

blackfennec_doubles.extension_system.double_extensions_descriptor, 155

blackfennec_doubles.extension_system.double_extensions_validator, 156

blackfennec_doubles.extension_system.double_extensions_validator, 156

blackfennec_doubles.extension_system.double_extensions_validator, 156

blackfennec_doubles.extension_system.extension_api_factory, 155

blackfennec_doubles.layers, 158

blackfennec_doubles.layers.double_layer, 158

blackfennec_doubles.layers.encapsulation_base, 158

blackfennec_doubles.layers.encapsulation_base.double_factory, 158

blackfennec_doubles.layers.observable, 158

blackfennec_doubles.layers.observable.double_observable, 158

blackfennec_doubles.presentation_system, 163

blackfennec_doubles.presentation_system.double_history_serializer, 162

blackfennec_doubles.presentation_system.double_presenter_factory, 162

blackfennec_doubles.presentation_system.double_presenter_registry, 162

blackfennec_doubles.presentation_system.double_presenter_validator, 162

blackfennec_doubles.presentation_system.double_resource_type, 162

blackfennec_doubles.presentation_system.double_structure_serializer, 162

blackfennec_doubles.presentation_system.double_structure_serializer, 162

blackfennec_doubles.presentation_system.double_view_factor, 162

blackfennec_doubles.presentation_system.double_view_factor, 162

blackfennec_doubles.presentation_system.main_window	blackfennec_doubles.type_system.interpretation.double_coverage
159	166
blackfennec_doubles.presentation_system.main_window.document_type	blackfennec_doubles.type_system.interpretation.double_inte
159	166
blackfennec_doubles.presentation_system.navigation_service	blackfennec_doubles.type_system.interpretation.double_inte
160	167
blackfennec_doubles.presentation_system.navigation_service_double_type_system_service	blackfennec_doubles.type_system.interpretation.double_offe
159	167
blackfennec_doubles.presentation_system.ui_service	blackfennec_doubles.type_system.interpretation.double_spec
161	167
blackfennec_doubles.presentation_system.ui_service_double_message_util	blackfennec_doubles.util, 170
160	blackfennec_doubles.util.double_change_notification,
blackfennec_doubles.presentation_system.ui_service_double_message_overlay	blackfennec_doubles.util.double_comparable,
160	blackfennec_doubles.util.double_comparable,
blackfennec_doubles.presentation_system.ui_service_double_ui_service	blackfennec_doubles.util.double_meta_info,
160	blackfennec_doubles.util.double_meta_info,
blackfennec_doubles.presentation_system.ui_service_double_ui_service_registry	blackfennec_doubles.util.double_observable,
160	blackfennec_doubles.util.double_observable,
blackfennec_doubles.structure, 166	170
blackfennec_doubles.structure.double_boolean	blackfennec_doubles.util.double_service_locator,
163	170
blackfennec_doubles.structure.double_list,	t
163	tests, 214
blackfennec_doubles.structure.double_map, 164	tests.blackfennec, 212
blackfennec_doubles.structure.double_null,	tests.blackfennec.action_system, 171
164	tests.blackfennec.action_system.test_action,
blackfennec_doubles.structure.double_number,	tests.blackfennec.action_system.test_action_registry,
164	171
blackfennec_doubles.structure.double_reference	tests.blackfennec.action_system.test_context,
164	171
blackfennec_doubles.structure.double_root,	tests.blackfennec.document_system, 177
165	tests.blackfennec.document_system.mime_type,
blackfennec_doubles.structure.double_string,	174
165	tests.blackfennec.document_system.mime_type.json,
blackfennec_doubles.structure.double_structure,	174
165	tests.blackfennec.document_system.mime_type.json.test_json
blackfennec_doubles.structure.reference_navigation,	174
163	tests.blackfennec.document_system.mime_type.json.test_json
blackfennec_doubles.structure.reference_navigation_double_navigation	174
163	tests.blackfennec.document_system.mime_type.json.test_json
blackfennec_doubles.type_system, 169	172
blackfennec_doubles.type_system.double_type,	tests.blackfennec.document_system.mime_type.json.test_json
167	173
blackfennec_doubles.type_system.double_type_loader	tests.blackfennec.document_system.mime_type.json.test_stru
168	173
blackfennec_doubles.type_system.double_type_parser	tests.blackfennec.document_system.mime_type.test_mime_type
168	174
blackfennec_doubles.type_system.double_type_registry	tests.blackfennec.document_system.mime_type.test_mime_type
168	174
blackfennec_doubles.type_system.interpretation	tests.blackfennec.document_system.resource_type,
167	176
blackfennec_doubles.type_system.interpretation_double_action_document_system_resource_type_protocols	tests.blackfennec.document_system.resource_type.protocols,
166	175

tests.blackfennec.document_system.resource_type_protocols	174	tests.blackfennec.layers.merge_type_merged_phantom,	187
tests.blackfennec.document_system.resource_type_protocols	175	tests.blackfennec.layers.merge_type_merged_phantom,	187
tests.blackfennec.document_system.resource_type_protocols	175	tests.blackfennec.layers.observable,	190
tests.blackfennec.document_system.resource_type_protocols	175	tests.blackfennec.layers.observable.test_integration_observation,	188
tests.blackfennec.document_system.test_document,	176	tests.blackfennec.layers.observable.test_observation,	189
tests.blackfennec.document_system.test_document_factory,	176	tests.blackfennec.layers.observable.test_observation_base,	189
tests.blackfennec.document_system.test_document_registry,	177	tests.blackfennec.layers.observable.test_observation_entry,	189
tests.blackfennec.document_system.test_document_registry,	177	tests.blackfennec.layers.observable.test_observation_factory,	190
tests.blackfennec.extension_system,	181	tests.blackfennec.layers.overlay,	191
tests.blackfennec.extension_system.test_extensions,	177	tests.blackfennec.layers.overlay.test_integration_overlay,	191
tests.blackfennec.extension_system.test_extensions_api,	177	tests.blackfennec.layers.overlay.test_overlay,	190
tests.blackfennec.extension_system.test_extensions_registry,	179	tests.blackfennec.layers.overlay.test_overlay_base,	190
tests.blackfennec.extension_system.test_extensions_service,	179	tests.blackfennec.layers.overlay.test_overlay_factory_visitor,	190
tests.blackfennec.extension_system.test_presenter_registry,	180	tests.blackfennec.presentation_system,	196
tests.blackfennec.extension_system.test_view_factory,	180	tests.blackfennec.presentation_system.main_window,	194
tests.blackfennec.extension_system.test_view_factory,	180	tests.blackfennec.presentation_system.main_window.test_blackfennec,	191
tests.blackfennec.layers,	191	tests.blackfennec.presentation_system.main_window.test_document,	193
tests.blackfennec.layers.encapsulation_base,	185	tests.blackfennec.presentation_system.navigation_service,	194
tests.blackfennec.layers.encapsulation_base.test_base_factory_visitor,	181	tests.blackfennec.presentation_system.navigation_service.test_blackfennec,	194
tests.blackfennec.layers.encapsulation_base.test_encapsulation_base,	182	tests.blackfennec.presentation_system.navigation_service.test_blackfennec,	194
tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base,	183	tests.blackfennec.presentation_system.test_history_service,	194
tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base,	184	tests.blackfennec.presentation_system.ui_service,	196
tests.blackfennec.layers.encapsulation_base.test_reference_encapsulation_base,	185	tests.blackfennec.presentation_system.ui_service.test_message,	196
tests.blackfennec.layers.merge,	188	tests.blackfennec.presentation_system.ui_service.test_ui_service,	195
tests.blackfennec.layers.merge.test_integration_merged_list,	185	tests.blackfennec.structure,	203
tests.blackfennec.layers.merge.test_merged_list,	187	tests.blackfennec.structure.reference_navigation,	199
tests.blackfennec.layers.merge.test_merged_map,	187	tests.blackfennec.structure.reference_navigation.test_child,	196
tests.blackfennec.layers.merge.test_merged_null,	187	tests.blackfennec.structure.reference_navigation.test_index,	197
		tests.blackfennec.structure.reference_navigation.test_navigation,	197

[tests.blackfennec.structure.reference_navigation](#), 198
[tests.blackfennec.structure.reference_navigation](#), 198
[tests.blackfennec.structure.reference_navigation](#), 198
[tests.blackfennec.structure.reference_navigation](#), 199
[tests.blackfennec.structure.test_boolean](#), 199
[tests.blackfennec.structure.test_list](#), 200
[tests.blackfennec.structure.test_map](#), 200
[tests.blackfennec.structure.test_null](#), 201
[tests.blackfennec.structure.test_number](#), 201
[tests.blackfennec.structure.test_reference](#), 202
[tests.blackfennec.structure.test_string](#), 202
[tests.blackfennec.structure.test_structure](#), 202
[tests.blackfennec.type_system](#), 209
[tests.blackfennec.type_system.interpretation](#), 205
[tests.blackfennec.type_system.interpretation.test_coverage](#), 203
[tests.blackfennec.type_system.interpretation.test_integration_interpretation_service](#), 203
[tests.blackfennec.type_system.interpretation.test_interpretation](#), 203
[tests.blackfennec.type_system.interpretation.test_interpretation_service](#), 204
[tests.blackfennec.type_system.interpretation.test_offer](#), 204
[tests.blackfennec.type_system.interpretation.test_specification](#), 205
[tests.blackfennec.type_system.test_boolean_type](#), 205
[tests.blackfennec.type_system.test_integration_type_loader](#), 205
[tests.blackfennec.type_system.test_list_type](#), 206
[tests.blackfennec.type_system.test_map_type](#), 207
[tests.blackfennec.type_system.test_null_type](#), 207
[tests.blackfennec.type_system.test_number_type](#), 208
[tests.blackfennec.type_system.test_string_type](#), 208
[tests.blackfennec.type_system.test_type_base](#), 209
[tests.blackfennec.type_system.test_type_parser](#), 209
[tests.blackfennec.type_system.test_type_registry](#), 209
[tests.blackfennec.util](#), 212
[tests.blackfennec.util.test_change_notification](#), 209
[tests.blackfennec.util.test_change_notification_dispatch](#), 210
[tests.blackfennec.util.test_comparable](#), 210
[tests.blackfennec.util.test_deep_copy](#), 210
[tests.blackfennec.util.test_integration_service_locator](#), 211
[tests.blackfennec.util.test_meta_info](#), 211
[tests.blackfennec.util.test_observable](#), 211
[tests.context](#), 214
[tests.test_utils](#), 214
[tests.test_utils.connection](#), 212
[tests.test_utils.deep_compare](#), 212
[tests.test_utils.observer](#), 214
[tests.test_utils.parameterize](#), 214
[tests.test_utils.what_the_farmer_does_not_eat_visitor_factory](#), 214
[tests.test_blackfennec_visitor](#), 209
[tests.test_blackfennec_visitor.test_change_notification](#), 209
[tests.test_blackfennec_visitor.test_change_notification_dispatch](#), 210
[tests.test_blackfennec_visitor.test_comparable](#), 210
[tests.test_blackfennec_visitor.test_deep_copy](#), 210
[tests.test_blackfennec_visitor.test_integration_service_locator](#), 211
[tests.test_blackfennec_visitor.test_meta_info](#), 211
[tests.test_blackfennec_visitor.test_observable](#), 211
[tests.test_blackfennec_visitor.test_utils](#), 214
[tests.test_blackfennec_visitor.connection](#), 212
[tests.test_blackfennec_visitor.deep_compare](#), 212
[tests.test_blackfennec_visitor.observer](#), 214
[tests.test_blackfennec_visitor.parameterize](#), 214
[tests.test_blackfennec_visitor.what_the_farmer_does_not_eat_visitor_factory](#), 214

INDEX

Symbols

- `_coverage` (*blackfennec.type_system.interpretation.offer.Offer attribute*), 139
 - `_mime_types` (*blackfennec.document_system.mime_type.mime_type_registry.MimeTypeRegistry attribute*), 109
 - `_presenter` (*blackfennec.presentation_system.navigation_service.navigation_service.NavigationService attribute*), 127
 - `_presenters` (*blackfennec.presentation_system.presenter_registry.PresenterRegistry attribute*), 129
 - `_resource_types` (*blackfennec.document_system.resource_type.resource_type_registry.ResourceTypeRegistry attribute*), 112
 - `_specificity` (*blackfennec.type_system.interpretation.offer.Offer attribute*), 139
 - `_subject` (*blackfennec.type_system.interpretation.offer.Offer attribute*), 139
 - `_type` (*blackfennec.type_system.interpretation.offer.Offer attribute*), 139
 - `_type_view_factories` (*blackfennec.presentation_system.type_view_factory_registry.TypeViewFactoryRegistry attribute*), 130
 - `_types` (*blackfennec.type_system.type_registry.TypeRegistry attribute*), 145
 - `_view_factory` (*blackfennec.type_system.interpretation.offer.Offer attribute*), 139
- A**
- `AboutWindowView` (class in *blackfennec.presentation_system.about_window.about_window_view*), 124
 - `AboutWindowViewModel` (class in *blackfennec.presentation_system.about_window.about_window_view_model*), 124
 - `ABSOLUTE_POINTER_PATTERN` (*blackfennec.document_system.mime_type.json.json_pointer_serializer.JsonPointerSerializer attribute*), 106
 - `ABSOLUTE_POINTER_PATTERN` (*blackfennec.document_system.mime_type.json.json_reference_serializer.JsonReferenceSerializer attribute*), 107
 - `accept()` (*blackfennec.layers.encapsulation_base.encapsulation_base.EncapsulationBase method*), 117
 - `accept()` (*blackfennec.layers.merge.merged_null.MergedNull method*), 120
 - `accept()` (*blackfennec.layers.merge.merged_phantom.MergedPhantom method*), 120
 - `accept()` (*blackfennec.layers.merge.merged_structure.MergedStructure method*), 120
 - `accept()` (*blackfennec.structure.boolean.Boolean method*), 133
 - `accept()` (*blackfennec.structure.list.List method*), 133
 - `accept()` (*blackfennec.structure.map.Map method*), 134
 - `accept()` (*blackfennec.structure.null.Null method*), 135
 - `accept()` (*blackfennec.structure.number.Number method*), 135
 - `accept()` (*blackfennec.structure.reference.Reference method*), 135
 - `accept()` (*blackfennec.structure.string.String method*), 136
 - `accept()` (*blackfennec.structure.structure.Structure method*), 136
 - `accept()` (*blackfennec_doubles.structure.double_boolean.BooleanMock method*), 163
 - `accept()` (*blackfennec_doubles.structure.double_list.ListMock method*), 163
 - `accept()` (*blackfennec_doubles.structure.double_map.MapMock method*), 164
 - `accept()` (*blackfennec_doubles.structure.double_null.NullMock method*), 164
 - `accept()` (*blackfennec_doubles.structure.double_number.NumberMock method*), 164
 - `accept()` (*blackfennec_doubles.structure.double_reference.ReferenceMock method*), 164
 - `accept()` (*blackfennec_doubles.structure.double_root.RootMock method*), 165
 - `accept()` (*blackfennec_doubles.structure.double_string.StringMock method*), 165
 - `accept()` (*blackfennec_doubles.structure.double_structure.StructureInstance method*), 165
 - `accept()` (*blackfennec_doubles.structure.double_structure.StructureMock method*), 165

method), 165

Action (class in *blackfennec.action_system.action*), 104

Action (class in *tests.blackfennec.layers.observable.test_integration_observable_layer*), 188

Action (class in *tests.blackfennec.presentation_system.test_history_service*), 195

action_name (blackfennec.presentation_system.ui_service.message.Message property), 128

action_registry (blackfennec.extension_system.extension_api.ExtensionApi property), 115

action_registry() (in module *tests.blackfennec.extension_system.test_extension_service*), 177

action_target (blackfennec.presentation_system.ui_service.message.Message property), 128

ActionMock (class in *blackfennec_doubles.action_system.double_action*), 149

ActionRegistry (class in *blackfennec.action_system.action_registry*), 104

ActionRegistryMock (class in *blackfennec_doubles.action_system.double_action_registry*), 149

actions (*blackfennec.action_system.action_registry.ActionRegistry* property), 104

activate() (blackfennec.extension_system.extension.Extension method), 115

activate() (blackfennec_doubles.extension_system.double_extension.ExtensionMock method), 156

ACTIVE (*blackfennec.extension_system.extension.Extension* attribute), 114

add_element() (blackfennec.type_system.list_type.ListType method), 141

ADD_ITEM (*tests.blackfennec.layers.observable.test_integration_observable_layer.Action* attribute), 188

add_item() (blackfennec.layers.encapsulation_base.list_encapsulation_base.ListEncapsulationBase method), 118

add_item() (blackfennec.layers.encapsulation_base.map_encapsulation_base.MapEncapsulationBase method), 118

add_item() (*blackfennec.structure.list.List* method), 133

add_item() (*blackfennec.structure.map.Map* method), 134

add_item() (blackfennec_doubles.structure.double_list.ListMock method), 163

add_item() (blackfennec_doubles.structure.double_map.MapMock method), 164

add_to_history() (blackfennec_doubles.presentation_system.double_history_service.History method), 161

add_toast() (blackfennec_doubles.presentation_system.ui_service.double_message_overlay.MessageOverlay method), 160

api() (in module *tests.blackfennec.extension_system.test_extension_service*), 179

apply() (*blackfennec.layers.merge.merged_layer.MergedLayer* method), 119

apply() (*blackfennec.layers.observable.observable.ObservableLayer* method), 122

apply() (*blackfennec.layers.overlay.overlay.Overlay* method), 123

apply() (*blackfennec_doubles.layers.double_layer.LayerMock* method), 158

assert_state() (blackfennec_doubles.extension_system.double_extension.ExtensionMock method), 156

attach_tab() (blackfennec.presentation_system.main_window.black_fennec_view_model.BlackFennecViewModel method), 124

auction() (*blackfennec_doubles.type_system.interpretation.double_auctioneer.Auctioneer* method), 166

AuctioneerMock (class in *blackfennec_doubles.type_system.interpretation.double_auctioneer*), 166

B

BaseFactoryVisitor (class in *blackfennec.layers.encapsulation_base.base_factory_visitor*), 116

BaseFactoryVisitorTestSuite (class in *tests.blackfennec.layers.encapsulation_base.test_base_factory_visitor*), 116

BFieldTypeResourceType (class in *blackfennec.document_system.resource_type.protocols.bftype_resource_type*), 110

EncapsulationBase (class in *blackfennec.layers.encapsulation_base.encapsulation_base*), 117

bind() (*blackfennec.layers.encapsulation_base.encapsulation_base.EncapsulationBase* method), 117

bind() (*blackfennec.layers.encapsulation_base.map_encapsulation_base.MapEncapsulationBase* method), 148

blackfennec module, 149

blackfennec.action_system module, 105

blackfennec.action_system.action module, 104

blackfennec.action_system.action_registry module, 104

module, 104	module, 123
blackfennec.action_system.context	blackfennec.layers.encapsulation_base
module, 104	module, 119
blackfennec.document_system	blackfennec.layers.encapsulation_base.base_factory_visitor
module, 114	module, 116
blackfennec.document_system.document	blackfennec.layers.encapsulation_base.encapsulation_base
module, 113	module, 117
blackfennec.document_system.document_factory	blackfennec.layers.encapsulation_base.list_encapsulation_base
module, 114	module, 118
blackfennec.document_system.document_registry	blackfennec.layers.encapsulation_base.map_encapsulation_base
module, 114	module, 118
blackfennec.document_system.mime_type	blackfennec.layers.encapsulation_base.reference_encapsulation_base
module, 110	module, 119
blackfennec.document_system.mime_type.in_memory	blackfennec.layers.merge
module, 105	module, 122
blackfennec.document_system.mime_type.in_memory.in_memory_mime_type	blackfennec.layers.merge.merged_layer
module, 105	module, 119
blackfennec.document_system.mime_type.json	blackfennec.layers.merge.merged_list
module, 108	module, 120
blackfennec.document_system.mime_type.json.json_mime_type	blackfennec.layers.merge.merged_map
module, 105	module, 120
blackfennec.document_system.mime_type.json.json_mime_type.serialize	blackfennec.layers.merge.merged_null
module, 106	module, 120
blackfennec.document_system.mime_type.json.json_mime_type.serialize_in_memory	blackfennec.layers.merge.merged_phantom
module, 107	module, 120
blackfennec.document_system.mime_type.mime_type_registry	blackfennec.layers.merge.merged_structure
module, 108	module, 120
blackfennec.document_system.mime_type.mime_type_registry	blackfennec.layers.merge.merger
module, 109	module, 121
blackfennec.document_system.resource_type	blackfennec.layers.observable
module, 113	module, 123
blackfennec.document_system.resource_type.prototype	blackfennec.layers.observable.observable
module, 112	module, 122
blackfennec.document_system.resource_type.prototype.layer_resource_type	blackfennec.layers.observable.observable_base
module, 110	module, 122
blackfennec.document_system.resource_type.prototype.layer_resource_type	blackfennec.layers.observable.observable_factory_visitor
module, 110	module, 123
blackfennec.document_system.resource_type.prototype.layer_resource_type	blackfennec.layers.overlay.overlay
module, 111	module, 123
blackfennec.document_system.resource_type.resource_registry	blackfennec.layers.overlay.overlay_base
module, 112	module, 123
blackfennec.extension_system	blackfennec.layers.overlay.overlay_factory_visitor
module, 116	module, 123
blackfennec.extension_system.extension	blackfennec.presentation_system
module, 114	module, 131
blackfennec.extension_system.extension_api	blackfennec.presentation_system.about_window
module, 115	module, 124
blackfennec.extension_system.extension_registry	blackfennec.presentation_system.about_window.about_window
module, 116	module, 124
blackfennec.extension_system.extension_service	blackfennec.presentation_system.about_window.about_window
module, 116	module, 124
blackfennec.layers	blackfennec.presentation_system.extension_warning_dialog

module, 124
 blackfennec.presentation_system.extension_warning_dialog
 module, 124
 blackfennec.presentation_system.history_service
 module, 128
 blackfennec.presentation_system.main_window
 module, 126
 blackfennec.presentation_system.main_window.blackfennec_view
 module, 124
 blackfennec.presentation_system.main_window.blackfennec_view_model
 module, 124
 blackfennec.presentation_system.main_window.dock_fennec_tab
 module, 125
 blackfennec.presentation_system.main_window.dock_fennec_view
 module, 126
 blackfennec.presentation_system.main_window.fill_color_picker
 module, 126
 blackfennec.presentation_system.navigation_service
 module, 127
 blackfennec.presentation_system.navigation_service.proxy
 module, 126
 blackfennec.presentation_system.navigation_service.service
 module, 127
 blackfennec.presentation_system.presenter_registry
 module, 129
 blackfennec.presentation_system.structure_view
 module, 129
 blackfennec.presentation_system.type_view
 module, 130
 blackfennec.presentation_system.type_view_factory
 module, 130
 blackfennec.presentation_system.type_view_factory.view
 module, 130
 blackfennec.presentation_system.ui_service
 module, 128
 blackfennec.presentation_system.ui_service.message
 module, 128
 blackfennec.presentation_system.ui_service.mini_type_selector
 module, 127
 blackfennec.presentation_system.ui_service.mini_type_selector.view
 module, 127
 blackfennec.presentation_system.ui_service.ui_service
 module, 128
 blackfennec.structure
 module, 137
 blackfennec.structure.boolean
 module, 133
 blackfennec.structure.list
 module, 133
 blackfennec.structure.map
 module, 134
 blackfennec.structure.null
 module, 135
 blackfennec.structure.number
 module, 135
 blackfennec.structure.reference_navigation
 module, 133
 blackfennec.structure.reference_navigation.child_navigator
 module, 131
 blackfennec.structure.reference_navigation.index_of_navigator
 module, 131
 blackfennec.structure.reference_navigation.navigator
 module, 132
 blackfennec.structure.reference_navigation.parent_navigator
 module, 132
 blackfennec.structure.reference_navigation.root_navigator
 module, 132
 blackfennec.structure.reference_navigation.sibling_offset
 module, 132
 blackfennec.structure.reference_navigation.uri_navigator
 module, 133
 blackfennec.structure.string
 module, 136
 blackfennec.structure.structure_serializer
 module, 136
 blackfennec.structure.visitor
 module, 137
 blackfennec.type_system
 module, 146
 blackfennec.type_system.boolean_type
 module, 141
 blackfennec.type_system.interpretation
 module, 141
 blackfennec.type_system.interpretation.coverage
 module, 137
 blackfennec.type_system.interpretation.interpretation
 module, 138
 blackfennec.type_system.interpretation.interpretation_service
 module, 139
 blackfennec.type_system.interpretation.interpretation_view
 module, 139
 blackfennec.type_system.interpretation.specification
 module, 140
 blackfennec.type_system.list_type
 module, 141
 blackfennec.type_system.map_type
 module, 142
 blackfennec.type_system.null_type
 module, 142
 blackfennec.type_system.number_type
 module, 142
 blackfennec.type_system.reference_type
 module, 143
 blackfennec.type_system.string_type

module, 143
 blackfennec.type_system.type
 module, 143
 blackfennec.type_system.type_coverage_mixin
 module, 144
 blackfennec.type_system.type_factory
 module, 144
 blackfennec.type_system.type_loader
 module, 145
 blackfennec.type_system.type_parser
 module, 145
 blackfennec.type_system.type_registry
 module, 145
 blackfennec.util
 module, 149
 blackfennec.util.change_notification
 module, 146
 blackfennec.util.change_notification_dispatcher
 module, 146
 blackfennec.util.comparable
 module, 146
 blackfennec.util.deep_copy
 module, 146
 blackfennec.util.intercepting_visitor
 module, 147
 blackfennec.util.meta_info
 module, 147
 blackfennec.util.observable
 module, 148
 blackfennec.util.parameterized_visitor
 module, 148
 blackfennec.util.service_locator
 module, 148
 blackfennec.util.type_comparator
 module, 148
 blackfennec_doubles
 module, 170
 blackfennec_doubles.action_system
 module, 150
 blackfennec_doubles.action_system.double_action
 module, 149
 blackfennec_doubles.action_system.double_action_registry
 module, 149
 blackfennec_doubles.action_system.double_context
 module, 149
 blackfennec_doubles.action_system.double_ui_controller
 module, 150
 blackfennec_doubles.document_system
 module, 154
 blackfennec_doubles.document_system.double_document
 module, 154
 blackfennec_doubles.document_system.double_document_factory
 module, 154
 blackfennec_doubles.document_system.double_document_registry
 module, 154
 blackfennec_doubles.document_system.mime_type
 module, 153
 blackfennec_doubles.document_system.mime_type.double_mime
 module, 152
 blackfennec_doubles.document_system.mime_type.double_mime.
 module, 152
 blackfennec_doubles.document_system.mime_type.json
 module, 152
 blackfennec_doubles.document_system.mime_type.json.double_
 module, 150
 blackfennec_doubles.document_system.mime_type.json.double_
 module, 151
 blackfennec_doubles.document_system.mime_type.json.double_
 module, 151
 blackfennec_doubles.document_system.resource_type
 module, 154
 blackfennec_doubles.document_system.resource_type.double_r
 module, 153
 blackfennec_doubles.document_system.resource_type.double_r
 module, 153
 blackfennec_doubles.double_dummy
 module, 170
 blackfennec_doubles.extension_system
 module, 157
 blackfennec_doubles.extension_system.double_extension
 module, 156
 blackfennec_doubles.extension_system.double_extension_api
 module, 156
 blackfennec_doubles.extension_system.double_extension_load
 module, 156
 blackfennec_doubles.extension_system.double_extension_regi
 module, 157
 blackfennec_doubles.extension_system.double_extension_sour
 module, 157
 blackfennec_doubles.extension_system.double_extension_sour
 module, 157
 blackfennec_doubles.extension_system.double_extensions
 module, 156
 blackfennec_doubles.extension_system.double_extensions.cre
 module, 155
 blackfennec_doubles.extension_system.double_extensions.cre
 module, 155
 blackfennec_doubles.extension_system.double_extensions.des
 module, 155
 blackfennec_doubles.extension_system.double_extensions.des
 module, 155
 blackfennec_doubles.extension_system.double_extensions.val
 module, 156
 blackfennec_doubles.extension_system.double_extensions.val
 module, 155
 blackfennec_doubles.extension_system.extension_api_factory
 module, 157
 blackfennec_doubles.layers

module, 158	module, 163
blackfennec_doubles.layers.double_layer	blackfennec_doubles.structure.double_list
module, 158	module, 163
blackfennec_doubles.layers.encapsulation_base	blackfennec_doubles.structure.double_map
module, 158	module, 164
blackfennec_doubles.layers.encapsulation_base	blackfennec_doubles.structure.double_null
module, 157	module, 164
blackfennec_doubles.layers.observable	blackfennec_doubles.structure.double_number
module, 158	module, 164
blackfennec_doubles.layers.observable.double_observable	blackfennec_doubles.structure.double_reference
module, 158	module, 164
blackfennec_doubles.presentation_system	blackfennec_doubles.structure.double_root
module, 163	module, 165
blackfennec_doubles.presentation_system.double_boolean	blackfennec_doubles.structure.double_string
module, 161	module, 165
blackfennec_doubles.presentation_system.double_factorials	blackfennec_doubles.structure.double_structure
module, 161	module, 165
blackfennec_doubles.presentation_system.double_factors	blackfennec_doubles.structure.reference_navigation
module, 161	module, 163
blackfennec_doubles.presentation_system.double_double	blackfennec_doubles.structure.reference_navigation.double_
module, 161	module, 163
blackfennec_doubles.presentation_system.double_double_list	blackfennec_doubles.type_system
module, 162	module, 169
blackfennec_doubles.presentation_system.double_double_list	blackfennec_doubles.type_system.double_type
module, 162	module, 167
blackfennec_doubles.presentation_system.double_double_loader	blackfennec_doubles.type_system.double_type_loader
module, 162	module, 168
blackfennec_doubles.presentation_system.double_double_loader	blackfennec_doubles.type_system.double_type_parser
module, 162	module, 168
blackfennec_doubles.presentation_system.double_double_loader	blackfennec_doubles.type_system.double_type_registry
module, 162	module, 168
blackfennec_doubles.presentation_system.double_double_loader	blackfennec_doubles.type_system.interpretation
module, 162	module, 167
blackfennec_doubles.presentation_system.main_window	blackfennec_doubles.type_system.interpretation.double_auct
module, 159	module, 166
blackfennec_doubles.presentation_system.main_window	blackfennec_doubles.type_system.interpretation.double_cove
module, 159	module, 166
blackfennec_doubles.presentation_system.navigation	blackfennec_doubles.type_system.interpretation.double_inte
module, 160	module, 166
blackfennec_doubles.presentation_system.navigation	blackfennec_doubles.type_system.interpretation.double_inte
module, 159	module, 167
blackfennec_doubles.presentation_system.ui_service	blackfennec_doubles.type_system.interpretation.double_offe
module, 161	module, 167
blackfennec_doubles.presentation_system.ui_service	blackfennec_doubles.type_system.interpretation.double_spec
module, 160	module, 167
blackfennec_doubles.presentation_system.ui_service	blackfennec_doubles.util.double_change_notification
module, 160	module, 170
blackfennec_doubles.presentation_system.ui_service	blackfennec_doubles.util.double_change_notification
module, 160	module, 169
blackfennec_doubles.presentation_system.ui_service	blackfennec_doubles.util.double_change_notification
module, 160	module, 169
blackfennec_doubles.structure	blackfennec_doubles.util.double_meta_info
module, 166	module, 169
blackfennec_doubles.structure.double_boolean	blackfennec_doubles.util.double_observable

module, 170
 blackfennec_doubles.util.double_service_locator
 module, 170
 BlackFennecMetaInfo (class in blackfennec.util.meta_info), 147
 BlackFennecViewModel (class in blackfennec.presentation_system.main_window.black_fennec_view_model), 124
 Boolean (class in blackfennec.structure.boolean), 133
 BooleanComparator (class in tests.test_utils.deep_compare), 212
 BooleanMerger (class in blackfennec.layers.merge.merger), 121
 BooleanMock (class in blackfennec_doubles.structure.double_boolean), 163
 BooleanTestSuite (class in tests.blackfennec.structure.test_boolean), 199
 BooleanType (class in blackfennec.type_system.boolean_type), 141
C
 calculate_coverage() (blackfennec.type_system.type_coverage_mixin.TypeCoverageMixin method), 144
 calculate_coverage() (blackfennec_doubles.type_system.double_type.TypeMock method), 167
 calls (tests.test_utils.observer.Observer property), 214
 can_get_parent_parameters() (in module tests.blackfennec.layers.merge.test_merged_structure), 187
 can_get_structure_parameters() (in module tests.blackfennec.layers.merge.test_merged_structure), 187
 can_handle_uri() (blackfennec.presentation_system.main_window.black_fennec_view_model.method), 124
 can_redo() (blackfennec.presentation_system.history_service.HistoryService method), 128
 can_redo() (blackfennec_doubles.presentation_system.double_history_service.HistoryServiceMock method), 161
 can_undo() (blackfennec.presentation_system.history_service.HistoryService method), 128
 can_undo() (blackfennec_doubles.presentation_system.double_history_service.HistoryServiceMock method), 161
 CHANGE (tests.blackfennec.layers.observable.test_integration_observable_layer.Action attribute), 188
 CHANGE (tests.blackfennec.presentation_system.test_history_service.Action attribute), 195
 change_notification() (in module tests.blackfennec.util.test_change_notification), 209
 change_notification_dispatch_mixin() (in module tests.blackfennec.util.test_change_notification_dispatch_mixin), 210
 ChangeNotification (class in blackfennec.util.change_notification), 146
 ChangeNotificationDispatchMixin (class in blackfennec.util.change_notification_dispatch_mixin), 146
 ChangeNotificationMock (class in blackfennec_doubles.util.double_change_notification), 169
 child() (in module tests.blackfennec.structure.reference_navigation.test_parent_navigator), 198
 ChildNavigator (class in blackfennec.structure.reference_navigation.child_navigator), 131
 close_file() (blackfennec.presentation_system.main_window.black_fennec_view_model.method), 124
 Comparable (class in blackfennec.util.comparable), 146
 ComparableMock (class in blackfennec_doubles.util.double_comparable), 169
 ComparableTestSuite (class in tests.blackfennec.util.test_comparable), 210
 ComparatorFactory (class in tests.test_utils.deep_compare), 212
 ComparatorTemplate (class in tests.test_utils.deep_compare), 212
 compare() (blackfennec.util.type_comparator.TypeComparator static method), 148
 compare() (tests.test_utils.deep_compare.DeepCompare static method), 213
 ConcreteAction (class in tests.blackfennec.action_system.test_action), 170
 content (blackfennec.document_system.document.Document property), 113
 content (blackfennec_doubles.document_system.double_document.Document property), 154
 content() (tests.blackfennec.document_system.test_document.TestDocument method), 176
 Context (class in blackfennec.action_system.context), 104
 context() (in module tests.blackfennec.presentation_system.ui_service.test_ui_service), 195
 ContextMock (class in blackfennec_doubles.action_system.double_context), 165

default (*blackfenec.type_system.number_type.NumberType* property), 142
 default (*blackfenec.type_system.reference_type.ReferenceType* property), 143
 default (*blackfenec.type_system.string_type.StringType* property), 143
 default (*blackfenec.type_system.type.Type* property), 143
 default (*blackfenec.doubles.type_system.double_type.TypeMock* property), 167
 dependencies (*blackfenec.extension_system.extension.Extension* property), 115
 DEPENDENCY_MISSING (*blackfenec.extension_system.extension.Extension.State* attribute), 114
 deregister_action() (*blackfenec.action_system.action_registry.ActionRegistry* method), 104
 deregister_action() (*blackfenec.doubles.action_system.double_action_registry.ActionRegistryMock* method), 149
 deregister_actions() (*blackfenec.extension_system.extension.Extension* method), 115
 deregister_extension_source() (*blackfenec.doubles.extension_system.double_extension_source_registry.ExtensionSourceRegistryMock* method), 157
 deregister_message_overlay() (*blackfenec.presentation_system.ui_service.ui_service.UiService* method), 128
 deregister_mime_type() (*blackfenec.document_system.mime_type.mime_type_registry.MimeTypeRegistry* method), 109
 deregister_mime_type() (*blackfenec.doubles.document_system.mime_type.double_mime_type_registry.MimeTypeRegistryMock* method), 152
 deregister_presenter() (*blackfenec.presentation_system.presenter_registry.PresenterRegistry* method), 129
 deregister_presenter() (*blackfenec.doubles.presentation_system.double_presenter_registry.PresenterRegistryMock* method), 161
 deregister_presenters() (*blackfenec.extension_system.extension.Extension* method), 115
 deregister_resource_type() (*blackfenec.document_system.resource_type.resource_type_registry.ResourceTypeRegistry* method), 112
 deregister_resource_type() (*blackfenec.doubles.document_system.resource_type.double_resource_type_registry.ResourceTypeRegistryMock* method), 153
 deregister_type() (*blackfenec.type_system.type_registry.TypeRegistry* method), 145
 deregister_type() (*blackfenec.doubles.type_system.double_type_registry.TypeRegistryMock* method), 168
 deregister_type_view_factory() (*blackfenec.presentation_system.type_view_factory_registry.TypeViewFactoryRegistry* method), 130
 deregister_type_view_factory() (*blackfenec.doubles.presentation_system.double_view_factory_registry.TypeViewFactoryRegistryMock* method), 162
 deregister_types() (*blackfenec.extension_system.extension.Extension* method), 115
 deregister_view_factories() (*blackfenec.extension_system.extension.Extension* method), 115
 description (*blackfenec.action_system.action.Action* property), 104
 description (*tests.blackfenec.action_system.test_action.ConcreteAction* property), 170
 describe() (*blackfenec.document_system.mime_type.json.json_reference_serializer.JsonReferenceSerializer* method), 108
 deserialize() (*blackfenec.structure.structure_serializer.StructureSerializer* method), 136
 deserialize() (*blackfenec.doubles.extension_system.double_extension_source_registry.ExtensionSourceRegistryMock* method), 151
 deserialize() (*blackfenec.doubles.document_system.mime_type.json.double_json_reference_serializer.DoubleJsonReferenceSerializer* method), 151
 deserialize() (*blackfenec.doubles.document_system.mime_type.json.double_structure_serializer.DoubleStructureSerializer* method), 151
 deserialize_absolute_pointer() (*blackfenec.document_system.mime_type.json.json_pointer_serializer.JsonPointerSerializer* class method), 106
 deserialize_absolute_mime_pointer() (*blackfenec.doubles.document_system.mime_type.json.double_json_pointer_serializer.DoubleJsonPointerSerializer* method), 150
 deserialize_relative_pointer() (*blackfenec.document_system.mime_type.json.json_pointer_serializer.JsonPointerSerializer* class method), 106
 deserialize_presenter_registry() (*blackfenec.doubles.document_system.mime_type.json.double_json_pointer_serializer.DoubleJsonPointerSerializer* method), 150
 destroy_extension() (*blackfenec.doubles.extension_system.double_extensions.create_failing_extensions.CreateFailingExtensionsRegistry* method), 155
 destroy_extensions() (*blackfenec.doubles.extension_system.double_extensions.destroy_failing_extensions.DestroyFailingExtensionsRegistry* method), 155
 destroy_extensions() (*blackfenec.doubles.extension_system.double_extensions.valid_extensions.ValidExtensionsRegistry* method), 155
 DestroyFailingExtensionMock (class in *blackfenec.doubles.extension_system.double_extensions.create_failing_extensions*)

nec_doubles.extension_system.double_extensions.document_factory (class in *blackfennec.extension.double_extensions*), 155

detach_tab() (*blackfennec.presentation_system.main_window.black_fennec_document_tab.BlackFennecViewModel* module method), 125

Document (class in *blackfennec.document_system.document*), 113

document() (in module *tests.blackfennec.presentation_system.main_window.test_black_fennec_document_system.mime_type.json.test_json_reference_serializer*), 173

document() (in module *tests.blackfennec.presentation_system.main_window.test_black_fennec_view_model*), 191

document() (in module *tests.blackfennec.presentation_system.main_window.test_document_factory*), 193

document() (in module *tests.blackfennec.presentation_system.main_window.test_document_factory_mock*), 193

document() (in module *tests.blackfennec.presentation_system.main_window.test_document_mock*), 199

document() (*tests.blackfennec.document_system.test_document.TestDocumentSystem.document_registry*), 176

document_factory (*blackfennec.extension_system.extension_api.ExtensionApi* property), 115

document_factory() (in module *tests.blackfennec.document_system.mime_type.json.test_json_reference_serializer*), 173

document_factory() (in module *tests.blackfennec.document_system.test_document_factory*), 176

document_factory() (in module *tests.blackfennec.extension_system.test_extension_api*), 177

document_factory() (in module *tests.blackfennec.presentation_system.main_window.test_document_tab*), 193

document_factory() (in module *tests.blackfennec.structure.reference_navigation.test_uri_navigator*), 199

document_factory() (in module *tests.blackfennec.type_system.test_integration_type_loader*), 205

document_registry (*blackfennec.extension_system.extension_api.ExtensionApi* property), 115

document_registry() (in module *tests.blackfennec.document_system.test_document_factory*), 176

document_registry() (in module *tests.blackfennec.extension_system.test_extension_api*), 177

document_registry() (*tests.blackfennec.document_system.test_document.TestDocument* method), 176

document_tab() (*blackfennec.extension.double_extensions.document_tab*), 155

document_tab_parametrized() (in module *tests.blackfennec.presentation_system.main_window.test_document_factory*), 193

DocumentFactory (class in *blackfennec.document_system.document_factory*), 193

DocumentFactoryMock (class in *blackfennec_doubles.document_system.double_document_factory*), 193

DocumentMock (class in *blackfennec_doubles.document_system.double_document*), 199

DocumentRegistry (class in *blackfennec.document_system.document_registry*), 114

DocumentRegistryMock (class in *blackfennec_doubles.document_system.double_document_registry*), 154

DocumentTab (class in *blackfennec.presentation_system.main_window.document_tab*), 125

DocumentTabMock (class in *blackfennec_doubles.presentation_system.main_window.double_document_tab*), 159

DocumentTabView (class in *blackfennec.presentation_system.main_window.document_tab_view*), 126

Dummy (class in *blackfennec_doubles.double_dummy*), 193

E

elements_navigation() (*blackfennec.type_system.list_type.ListType* property), 141

EncapsulationBase (class in *blackfennec.layers.encapsulation_base.encapsulation_base*), 117

EncapsulationBaseTestSuite (class in *tests.blackfennec.layers.encapsulation_base.test_encapsulation_base*), 182

endpoint() (*tests.test_utils.observer.Observer* method), 214

execute() (*blackfennec.action_system.action.Action* method), 104

execute() (*blackfennec_doubles.action_system.double_action.ActionMock* method), 149

execute() (*tests.blackfennec.action_system.test_action.ConcreteAction* method), 170

expected (*blackfennec.type_system.boolean_type.BooleanType*), 170

property), 141

export_structure() (blackfen- ExtensionSourceRegistryMock (class in blackfen- nec.document_system.mime_type.in_memory.in_memory_mime_type.in_memory_mime_type.double_extension_source_registry method), 105 157

export_structure() (blackfen- F FactoryBaseVisitorMock (class in blackfen- nec.document_system.mime_type.json.json_mime_type.JsonMimeType FACTORIES (blackfenec.type_system.type_parser.TypeParser method), 106 attribute), 145

export_structure() (blackfen- Factory() (in module nec.document_system.mime_type.mime_type.MimeTypeFactory() (in module tests.blackfenec.extension_system.test_view_factory), method), 108 180

export_structure() (blackfen- factory().MimeTypeMock (in module nec_doubles.document_system.mime_type.double_factory() MimeTypesMimeTypeMock (in module tests.blackfenec.extension_system.test_view_factory_registry), method), 152 180

Extension (class in blackfen- FactoryBaseVisitorMock (class in blackfen- nec.extension_system.extension), 114 nec_doubles.layers.encapsulation_base.double_factory_base_visitor), 157

extension() (in module FAILED (blackfenec.extension_system.extension.Extension.State tests.blackfenec.extension_system.test_extension), 177 attribute), 114

Extension.State (class in blackfen- FileResourceType (class in blackfen- nec.extension_system.extension), 114 nec.document_system.resource_type.protocols.file_resource_type), 110

extension_api() (in module FileResourceTypeTestSuite (class in tests.blackfenec.extension_system.test_extension_api), 177 tests.blackfenec.document_system.resource_type.protocols.test_file_resource_type), 174

extension_registry() (in module fix_focus() (blackfen- tests.blackfenec.extension_system.test_extension_registry), 179 nec.presentation_system.ui_service.ui_service.UiService method), 191

extension_source_registry() (in module fix_focus() (blackfen- tests.blackfenec.presentation_system.main_window.test_blackfenec_view_model), 191 nec_doubles.presentation_system.ui_service.double_ui_service.UiService method), 191

extension_sources (blackfen- ExtensionSourceRegistryMock (class in blackfen- nec_doubles.extension_system.double_extension_source_registry), 157 property), 157

ExtensionApi (class in blackfen- G get_about_window_view_model() (blackfen- nec.extension_system.extension_api), 115 nec.presentation_system.main_window.black_fenec_view_model method), 125

ExtensionApiMock (class in blackfen- get_actions() (blackfen- nec_doubles.extension_system.double_extension_api), 156 nec.action_system.action_registry.ActionRegistry method), 104

ExtensionLoadingServiceMock (class in blackfen- get_actions() (blackfen- nec_doubles.extension_system.double_extension_loading_service), 156 nec_doubles.action_system.double_action_registry.ActionRegistry method), 149

ExtensionMock (class in blackfen- get_app_id() (blackfen- nec_doubles.extension_system.double_extension), 156 nec.util.meta_info.BlackFenecMetaInfo method), 147

ExtensionRegistry (class in blackfen- get_authors() (blackfen- nec.extension_system.extension_registry), 116 nec.util.meta_info.BlackFenecMetaInfo method), 147

ExtensionRegistryMock (class in blackfen- get_authors() (blackfen- nec_doubles.extension_system.double_extension_registry), 157 method), 147

ExtensionService (class in blackfen- get_authors() (blackfen- nec.extension_system.extension_service), 116 nec_doubles.util.double_meta_info.MetaInfoMock method), 169

ExtensionSourceMock (class in blackfen- get_clipboard_async() (blackfen- nec_doubles.extension_system.double_extension_source), 157 nec.presentation_system.ui_service.ui_service.UiService class method), 128

get_copy_right()	(blackfen- nec.util.meta_info.BlackFennecMetaInfo method), 147	nec.util.meta_info.BlackFennecMetaInfo method), 147
get_copy_right()	(blackfen- nec_doubles.util.double_meta_info.MetaInfoMock method), 169	get_issue_page() (blackfen- nec_doubles.util.double_meta_info.MetaInfoMock method), 169
get_description()	(blackfen- nec.util.meta_info.BlackFennecMetaInfo method), 147	get_license() (blackfen- nec.util.meta_info.BlackFennecMetaInfo method), 147
get_description()	(blackfen- nec_doubles.util.double_meta_info.MetaInfoMock method), 169	get_license() (blackfen- nec_doubles.util.double_meta_info.MetaInfoMock method), 169
get_document()	(blackfen- nec.document_system.document_factory.DocumentFactory method), 114	get_name() (blackfen- nec.util.meta_info.BlackFennecMetaInfo method), 147
get_document()	(blackfen- nec.document_system.document_registry.DocumentRegistry method), 114	get_name() (blackfen- nec_doubles.util.double_meta_info.MetaInfoMock method), 169
get_document()	(blackfen- nec_doubles.document_system.double_document_factory.DocumentFactory method), 154	get_release_notes() (blackfen- nec.util.meta_info.BlackFennecMetaInfo method), 147
get_document()	(blackfen- nec_doubles.action_system.double_ui_context.UiContextMock method), 154	get_root() (blackfen- nec_doubles.action_system.double_ui_context.UiContextMock method), 154
get_document()	(blackfen- nec_doubles.structure.double_root.RootMock method), 165	get_root() (blackfen- nec_doubles.presentation_system.ui_service.double_message_ov method), 160
get_extensions()	(blackfen- nec.extension_system.extension_registry.ExtensionRegistry method), 116	get_stats() (blackfen- nec_doubles.layers.double_layer.LayerMock method), 158
get_extensions()	(blackfen- nec_doubles.extension_system.double_extension_registry.ExtensionRegistry method), 157	get_stats() (blackfen- nec_doubles.layers.encapsulation_base.double_factory_base_vis method), 157
get_factory()	(blackfen- nec.presentation_system.type_view_factory_registry.TypeViewFactoryRegistry method), 131	get_structure_from_clipboard_async() (black- fennec.presentation_system.ui_service.ui_service.UiService method), 131
get_factory()	(blackfen- nec_doubles.presentation_system.double_view_factory_registry.TypeViewFactoryRegistryMock method), 162	get_structure_from_clipboard_async() (black- fennec_doubles.presentation_system.ui_service.double_ui_servic method), 160
get_home_page()	(blackfen- nec.util.meta_info.BlackFennecMetaInfo method), 147	get_summary() (blackfen- nec.util.meta_info.BlackFennecMetaInfo method), 147
get_home_page()	(blackfen- nec_doubles.util.double_meta_info.MetaInfoMock method), 169	get_summary() (blackfen- nec_doubles.util.double_meta_info.MetaInfoMock method), 169
get_icon_path()	(blackfen- nec.util.meta_info.BlackFennecMetaInfo method), 147	get_version() (blackfen- nec.util.meta_info.BlackFennecMetaInfo method), 147
get_icon_path()	(blackfen- nec_doubles.util.double_meta_info.MetaInfoMock method), 169	get_version() (blackfen- nec_doubles.util.double_meta_info.MetaInfoMock method), 169
get_id()	(blackfennec_doubles.util.double_meta_info.MetaInfoMock method), 169	guess_mime_type() (blackfen- nec.document_system.resource_type.protocols.bftype_resource_ty method), 110
get_issue_page()	(blackfen- nec.util.meta_info.BlackFennecMetaInfo method), 147	guess_mime_type() (blackfen- nec.util.meta_info.BlackFennecMetaInfo method), 147

nec.document_system.resource_type.protocols.https_resource_type.HttpsResourceType
method), 111

guess_mime_type() (*blackfen-*
nec.document_system.resource_type.resource_type.ResourceType
method), 112

H

has_internet_connection() (*in* *module*
tests.test_utils.connection), 212

historized() (*in* *module*
tests.blackfenec.layers.observable.test_integration_observable_layer),
 188

history() (*in* *module*
tests.blackfenec.layers.observable.test_integration_observable_layer),
 188

history() (*in* *module*
tests.blackfenec.layers.observable.test_observable_factory_test_util),
 190

history() (*in* *module*
tests.blackfenec.presentation_system.test_history_service),
 196

HistoryEntry (*class* *in* *blackfen-*
nec.presentation_system.history_service),
 128

HistoryService (*class* *in* *blackfen-*
nec.presentation_system.history_service),
 128

HistoryServiceMock (*class* *in* *blackfen-*
nec_doubles.presentation_system.double_history_service),
 161

HttpsResourceType (*class* *in* *blackfen-*
nec.document_system.resource_type.protocols.https_resource_type),
 111

HttpsResourceTypeTestSuite (*class* *in*
tests.blackfenec.document_system.resource_type.protocols.test_integration_https_resource_type),
 175

I

i_notify_observers (*blackfen-*
nec_doubles.util.double_observable.ObservableMock
property), 170

icon (*blackfenec.presentation_system.main_window.document_tab.DocumentTab*
property), 125

import_structure() (*blackfen-*
nec.document_system.mime_type.in_memory.in_memory_mime_type
method), 105

import_structure() (*blackfen-*
nec.document_system.mime_type.json.json_mime_type.JsonMimeType
method), 106

import_structure() (*blackfen-*
nec.document_system.mime_type.mime_type.MimeType
method), 108

import_structure() (*blackfen-*
nec_doubles.document_system.mime_type.double_mime_type.MimeTypeMock
method), 107

INDEX (*blackfenec.extension_system.extension.Extension.State*
attribute), 115

IndexOfNavigator (*class* *in* *blackfen-*
nec.structure.reference_navigation.index_of_navigator),
 131

InMemoryMimeType (*class* *in* *blackfen-*
nec.document_system.mime_type.in_memory.in_memory_mime_t
ype), 105

installed() (*blackfen-*
nec_doubles.extension_system.double_extension_loading_service
method), 156

InterceptingVisitor (*class* *in* *blackfen-*
nec.util.intercepting_visitor), 147

interpret() (*blackfen-*
nec.type_system.interpretation.interpretation_service.Interpretati
onService
method), 139

interpret() (*blackfen-*
nec_doubles.type_system.interpretation.double_interpretation_se
rvices
method), 167

Interpretation (*class* *in* *blackfen-*
nec.type_system.interpretation.interpretation),
 138

interpretation_service (*blackfen-*
nec.extension_system.extension_api.ExtensionApi
property), 116

interpretation_service() (*in* *module*
tests.blackfenec.extension_system.test_extension_api),
 178

interpretation_service() (*in* *module*
tests.blackfenec.presentation_system.main_window.test_black_fe
nec
test_integration_https_resource_type),
 191

InterpretationMock (*class* *in* *blackfen-*
nec_doubles.type_system.interpretation.double_interpretation),
 166

InterpretationService (*class* *in* *blackfen-*
nec.type_system.interpretation.interpretation_service),
 139

InterpretationServiceMock (*class* *in* *blackfen-*
nec_doubles.type_system.interpretation.double_interpretation_se
rvice
Mock), 167

InterpretationTestSuite (*class* *in*
tests.blackfenec.type_system.interpretation.test_interpretation),
 203

interpreter() (*in* *module*
tests.blackfenec.type_system.interpretation.test_interpretation_s
ervice
Mock
method), 204

is_absolute_json_pointer() (*blackfen-*
nec.document_system.mime_type.json.json_pointer_serializer.Json
PointerSerializer
class *method*), 107

is_active (*blackfenec.extension_system.extension.Extension*
property), 115

<code>property</code>), 115		<code>JsonPointerSerializerMock</code> (class in <code>blackfennec_doubles.document_system.mime_type.json.double_json_pointer_serializer_mock</code>), 150
<code>is_child_optional()</code> (blackfennec.type_system.list_type.ListType method), 141		<code>JsonReferenceSerializer</code> (class in <code>blackfennec.document_system.mime_type.json.json_reference_serializer</code>), 107
<code>is_child_optional()</code> (blackfennec.type_system.map_type.MapType method), 142		<code>JsonReferenceSerializerMock</code> (class in <code>blackfennec_doubles.document_system.mime_type.json.double_json_reference_serializer_mock</code>), 151
<code>is_covered()</code> (blackfennec.type_system.interpretation.coverage.Coverage method), 138		
<code>is_covered()</code> (blackfennec_doubles.type_system.interpretation.double_coverage_method), 166		L
<code>is_optional</code> (blackfennec.type_system.type.Type property), 143		<code>LargeFileUtilsMock</code> (class in <code>blackfennec_utils.observer.Observer</code> property), 214
<code>is_reference()</code> (blackfennec.document_system.mime_type.json.json_reference_serializer class method), 108		<code>layer()</code> (in module <code>tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base</code>), 183
<code>is_reference()</code> (blackfennec_doubles.document_system.mime_type.json.double_json_reference_serializer class method), 151		<code>layer()</code> (in module <code>tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base</code>), 184
<code>is_relative_json_pointer()</code> (blackfennec.document_system.mime_type.json.json_pointer_serializer class method), 107		<code>layer()</code> (in module <code>tests.blackfennec.layers.encapsulation_base.test_reference_encapsulation_base</code>), 185
<code>is_request_for_preview</code> (blackfennec.type_system.interpretation.specification.Specification property), 140		<code>layer()</code> (in module <code>tests.blackfennec.layers.observable.test_observable</code>), 189
<code>is_request_for_preview</code> (blackfennec_doubles.type_system.interpretation.double_specification_property), 167		<code>layer()</code> (in module <code>tests.blackfennec.layers.overlay.test_integration_overlay</code>), 190
		<code>LayerMock</code> (class in <code>blackfennec_doubles.layers.double_layer</code>), 158
		<code>List</code> (class in <code>blackfennec.structure.list</code>), 133
		<code>ListComparisonMock</code> (class in <code>blackfennec.structure.reference_navigation.test_chi</code>), 196
		<code>list()</code> (in module <code>tests.blackfennec.structure.reference_navigation.test_in</code>), 197
J		<code>list_encapsulation_base()</code> (in module <code>tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base</code>), 183
<code>json_file()</code> (in module <code>tests.blackfennec.document_system.mime_type.json.test_json_mime_type</code>), 172		<code>ListComparator</code> (class in <code>tests.test_utils.deep_compare</code>), 213
<code>json_mime_type()</code> (in module <code>tests.blackfennec.document_system.mime_type.json.test_json_mime_type</code>), 172		<code>ListEncapsulationBase</code> (class in <code>blackfennec.layers.encapsulation_base.list_encapsulation_base</code>), 118
<code>json_reference_parser()</code> (in module <code>tests.blackfennec.document_system.mime_type.json.test_json_reference_serializer</code>), 173		<code>ListInstanceMock</code> (class in <code>blackfennec_doubles.structure.double_list</code>), 163
<code>json_reference_serializer()</code> (in module <code>tests.blackfennec.document_system.mime_type.json.test_json_reference_serializer</code>), 173		<code>ListMerger</code> (class in <code>blackfennec.layers.merge.merger</code>), 121
<code>json_string()</code> (in module <code>tests.blackfennec.document_system.mime_type.json.test_json_mime_type</code>), 172		<code>ListMock</code> (class in <code>blackfennec_doubles.structure.double_list</code>), 163
<code>JsonMimeType</code> (class in <code>blackfennec.document_system.mime_type.json.json_mime_type</code>), 105		<code>ListType</code> (class in <code>blackfennec.type_system.list_type</code>), 141
<code>JsonPointerSerializer</code> (class in <code>blackfennec.document_system.mime_type.json.json_pointer_serializer</code>), 106		<code>ListTypeTestSuite</code> (class in <code>tests.blackfennec.type_system.test_list_type</code>), 206
		<code>load()</code> (blackfennec.extension_system.extension_service.ExtensionService class method), 116

load() (blackfennec.type_system.type_loader.TypeLoader method), 145

load() (blackfennec.doubles.extension_system.double_extension_loading_service.ExtensionLoadingServiceMock method), 156

load() (blackfennec.doubles.type_system.double_type_loader.TypeLoaderMock method), 168

load_document() (blackfennec.presentation_system.main_window.document_document_loader.DocumentLoader method), 125

load_resource() (blackfennec.document_system.resource_type.protocols.bf_resource_type.ResourceType method), 110

load_resource() (blackfennec.document_system.resource_type.protocols.file_resource_type.FileResourceType method), 110

load_resource() (blackfennec.document_system.resource_type.protocols.https_resource_type.HttpsResourceType method), 111

load_resource() (blackfennec.document_system.resource_type.resource_type.ResourceType method), 112

load_resource() (blackfennec.doubles.document_system.resource_type.double_resource_type.DoubleResourceType method), 153

location (blackfennec.document_system.document.Document property), 113

M

Map (class in blackfennec.structure.map), 134

map() (in module tests.blackfennec.structure.reference_navigation.test_child_navigator), 196

map() (in module tests.blackfennec.structure.reference_navigation.test_index_of_navigator), 197

map_chain() (in module tests.blackfennec.layers.merge.test_integration_merge), 185

map_encapsulation_base() (in module tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base), 184

MapComparator (class in tests.test_utils.deep_compare), 213

MapEncapsulationBase (class in blackfennec.layers.encapsulation_base.map_encapsulation_base), 118

MapInstanceMock (class in blackfennec.doubles.structure.double_map), 164

MapMerger (class in blackfennec.layers.merge.merger), 121

MapMock (class in blackfennec.doubles.structure.double_map), 164

MapType (class in blackfennec.type_system.map_type), 142

MapTypeTestSuite (class in tests.blackfennec.type_system.test_map_type), 142

maximum (blackfennec.type_system.number_type.NumberType property), 142

merge() (blackfennec.layers.merge.merger.Merger method), 121

merged() (in module tests.blackfennec.layers.merge.test_merged_layer), 186

MergedLayerTab (class in blackfennec.layers.merge.merged_layer), 119

MergedList (class in blackfennec.layers.merge.merger), 120

MergedMap (class in blackfennec.layers.merge.merged_map), 120

MergedNull (class in blackfennec.layers.merge.merged_null), 120

MergedPhantom (class in blackfennec.layers.merge.merged_phantom), 120

MergedStructure (class in blackfennec.layers.merge.merged_structure), 120

Merger (class in blackfennec.layers.merge.merger), 121

MergerFactory (class in blackfennec.layers.merge.merger), 121

Message (class in blackfennec.presentation_system.ui_service.message), 194

message() (in module tests.blackfennec.presentation_system.ui_service.test_message), 194

message() (in module tests.blackfennec.presentation_system.ui_service.test_ui_service), 195

message_overlay() (in module tests.blackfennec.presentation_system.ui_service.test_ui_service), 195

MessageMock (class in blackfennec.doubles.presentation_system.ui_service.double_message), 160

MessageOverlayMock (class in blackfennec.doubles.presentation_system.ui_service.double_message_overlay), 160

metadata_storage (blackfennec.doubles.layers.encapsulation_base.double_factory_base_vis property), 157

metadata_storage (blackfennec.doubles.type_system.double_type_parser.TypeParserMock property), 168

MetaInfoMock (class in blackfennec.doubles.util.double_meta_info), 169

mime_type (blackfennec.document_system.document.Document property), 113

mime_type() (tests.blackfennec.document_system.test_document.TestDocument method), 176

mime_type_id (blackfennec.document_system.mime_type.in_memory.in_memory_mime_type property), 176

			blackfennec.document_system.document_factory,
mime_type_id	(blackfennec.document_system.mime_type.json.json_mime_type_property), 105	114	blackfennec.document_system.document_registry,
mime_type_id	(blackfennec.document_system.mime_type.mime_type.MimeType_property), 108	110	blackfennec.document_system.mime_type,
mime_type_id	(blackfennec.doubles.document_system.mime_type.double_mime_type_property), 152	105	blackfennec.document_system.mime_type.in_memory,
mime_type_registry	(blackfennec.extension_system.extension_api.ExtensionApi_property), 116	108	blackfennec.document_system.mime_type.in_memory.in_memory,
mime_type_registry()	(in module tests.blackfennec.document_system.test_document_factory), 176	105	blackfennec.document_system.mime_type.json,
mime_type_registry()	(in module tests.blackfennec.extension_system.test_extension_api), 178	107	blackfennec.document_system.mime_type.json.json_mime_type,
mime_types	(blackfennec.document_system.mime_type.mime_type_registry.MimeTypeRegistry_property), 109	109	blackfennec.document_system.mime_type.json.json_pointer,
mime_types	(blackfennec.doubles.document_system.mime_type.double_mime_type_registry.MimeTypeRegistryMock_property), 152	112	blackfennec.document_system.mime_type.mime_type,
MimeType	(class in blackfennec.document_system.mime_type.mime_type), 108	110	blackfennec.document_system.mime_type.mime_type_registry,
MimeTypeMock	(class in blackfennec.doubles.document_system.mime_type.double_mime_type), 152	110	blackfennec.document_system.resource_type,
MimeTypeRegistry	(class in blackfennec.document_system.mime_type.mime_type_registry), 109	111	blackfennec.document_system.resource_type.protocols,
MimeTypeRegistryMock	(class in blackfennec.doubles.document_system.mime_type.double_mime_type_registry), 152	111	blackfennec.document_system.resource_type.protocols.blackfennec,
MimeTypeRegistryTestSuite	(class in tests.blackfennec.document_system.mime_type.test_mime_type_registry), 174	112	blackfennec.document_system.resource_type.protocols.blackfennec.blackfennec,
MimeTypeTestSuite	(class in tests.blackfennec.document_system.mime_type.test_mime_type_registry), 174	112	blackfennec.document_system.resource_type.resource_type,
minimum	(blackfennec.type_system.number_type.NumberType_property), 142	116	blackfennec.document_system.resource_type.resource_type_registry,
module		116	blackfennec.document_system.resource_type.resource_type_registry,
blackfennec,	149	116	blackfennec.document_system.resource_type.resource_type_registry,
blackfennec.action_system,	105	116	blackfennec.document_system.resource_type.resource_type_registry,
blackfennec.action_system.action,	104	116	blackfennec.document_system.resource_type.resource_type_registry,
blackfennec.action_system.action_registry,	104	116	blackfennec.document_system.resource_type.resource_type_registry,
blackfennec.action_system.context,	104	116	blackfennec.document_system.resource_type.resource_type_registry,
blackfennec.document_system,	114	116	blackfennec.document_system.resource_type.resource_type_registry,
blackfennec.document_system.document,	113	116	blackfennec.document_system.resource_type.resource_type_registry,

blackfennec.layers.encapsulation_base.map_encapsulation_base, 118
 blackfennec.layers.encapsulation_base.reference_encapsulation_base, 119
 blackfennec.layers.merge, 122
 blackfennec.layers.merge.merged_layer, 119
 blackfennec.layers.merge.merged_list, 120
 blackfennec.layers.merge.merged_map, 120
 blackfennec.layers.merge.merged_null, 120
 blackfennec.layers.merge.merged_phantom, 120
 blackfennec.layers.merge.merged_structure, 120
 blackfennec.layers.merge.merger, 121
 blackfennec.layers.observable, 123
 blackfennec.layers.observable.observable, 122
 blackfennec.layers.observable.observable_base, 122
 blackfennec.layers.observable.observable_factory_visitor, 123
 blackfennec.layers.overlay, 123
 blackfennec.layers.overlay.overlay, 123
 blackfennec.layers.overlay.overlay_base, 123
 blackfennec.layers.overlay.overlay_factory_visitor, 123
 blackfennec.presentation_system, 131
 blackfennec.presentation_system.about_window, 124
 blackfennec.presentation_system.about_window.about_window_view_model, 124
 blackfennec.presentation_system.about_window.about_window_view, 124
 blackfennec.presentation_system.about_window.about_window_view_model, 124
 blackfennec.presentation_system.extension_warning_dialog, 124
 blackfennec.presentation_system.extension_warning_dialog.extension_warning_dialog, 124
 blackfennec.presentation_system.history_service, 128
 blackfennec.presentation_system.main_window, 126
 blackfennec.presentation_system.main_window.black_fennec_view, 124
 blackfennec.presentation_system.main_window.black_fennec_view_model, 124
 blackfennec.presentation_system.main_window.document_tab, 125
 blackfennec.presentation_system.main_window.document_tab.view, 126
 blackfennec.presentation_system.main_window.document_tab.view_model, 126
 blackfennec.presentation_system.main_window.file_column_view, 126
 blackfennec.presentation_system.navigation_service, 146
 blackfennec.presentation_system.navigation_service.navigation_service, 146
 blackfennec.presentation_system.navigation_service.navigation_service_model, 146
 blackfennec.presentation_system.presenter_registry, 129
 blackfennec.presentation_system.structure_view_factory, 129
 blackfennec.presentation_system.type_view, 130
 blackfennec.presentation_system.type_view_factory, 130
 blackfennec.presentation_system.type_view_factory_registry, 130
 blackfennec.presentation_system.ui_service, 128
 blackfennec.presentation_system.ui_service.message, 128
 blackfennec.presentation_system.ui_service.mime_type_service, 127
 blackfennec.presentation_system.ui_service.mime_type_service_model, 127
 blackfennec.presentation_system.ui_service.ui_service, 128
 blackfennec.structure, 137
 blackfennec.structure.boolean, 133
 blackfennec.structure.list, 133
 blackfennec.structure.map, 134
 blackfennec.structure.null, 135
 blackfennec.structure.number, 135
 blackfennec.structure.reference, 135
 blackfennec.structure.reference_navigation, 135
 blackfennec.structure.reference_navigation.child_navigation, 135
 blackfennec.structure.reference_navigation.index_of_name, 135
 blackfennec.structure.reference_navigation.navigator, 135
 blackfennec.structure.reference_navigation.parent_navigation, 135
 blackfennec.structure.reference_navigation.root_navigation, 135
 blackfennec.structure.reference_navigation.sibling_offset, 135
 blackfennec.structure.reference_navigation.uri_navigation, 135
 blackfennec.structure.string, 136
 blackfennec.structure.structure, 136
 blackfennec.structure.structure_serializer, 136
 blackfennec.structure.visitor, 137
 blackfennec.type_system, 146

blackfennec.type_system.boolean_type, 141	154
blackfennec.type_system.interpretation, 141	blackfennec_doubles.document_system.double_document_fa 154
blackfennec.type_system.interpretation.coverage, 137	blackfennec_doubles.document_system.double_document_re 154
blackfennec.type_system.interpretation.interpretation, 138	blackfennec_doubles.document_system.mime_type, 153
blackfennec.type_system.interpretation.interpretation.coverage, 139	blackfennec_doubles.document_system.mime_type.double_m 152
blackfennec.type_system.interpretation.offer, 139	blackfennec_doubles.document_system.mime_type.double_m 152
blackfennec.type_system.interpretation.specificity, 140	blackfennec_doubles.document_system.mime_type.json, 152
blackfennec.type_system.list_type, 141	blackfennec_doubles.document_system.mime_type.json.dou 150
blackfennec.type_system.map_type, 142	blackfennec_doubles.document_system.mime_type.json.dou 151
blackfennec.type_system.null_type, 142	blackfennec_doubles.document_system.mime_type.json.dou 151
blackfennec.type_system.number_type, 142	blackfennec_doubles.document_system.mime_type.json.dou 151
blackfennec.type_system.reference_type, 143	blackfennec_doubles.document_system.mime_type.json.dou 151
blackfennec.type_system.string_type, 143	blackfennec_doubles.document_system.resource_type, 154
blackfennec.type_system.type, 143	blackfennec_doubles.document_system.resource_type.dou 153
blackfennec.type_system.type_coverage_mixin, 144	blackfennec_doubles.document_system.resource_type.dou 153
blackfennec.type_system.type_factory, 144	blackfennec_doubles.document_system.resource_type.dou 153
blackfennec.type_system.type_loader, 145	blackfennec_doubles.double_dummy, 170
blackfennec.type_system.type_parser, 145	blackfennec_doubles.extension_system, 157
blackfennec.type_system.type_registry, 145	blackfennec_doubles.extension_system.double_extension, 156
blackfennec.util, 149	blackfennec_doubles.extension_system.double_extension_ 156
blackfennec.util.change_notification, 146	blackfennec_doubles.extension_system.double_extension_ 156
blackfennec.util.change_notification_dispatch_mixin, 146	blackfennec_doubles.extension_system.double_extension_ 157
blackfennec.util.comparable, 146	blackfennec_doubles.extension_system.double_extension_ 157
blackfennec.util.deep_copy, 146	blackfennec_doubles.extension_system.double_extension_ 157
blackfennec.util.intercepting_visitor, 147	blackfennec_doubles.extension_system.double_extensions 156
blackfennec.util.meta_info, 147	blackfennec_doubles.extension_system.double_extensions 155
blackfennec.util.observable, 148	blackfennec_doubles.extension_system.double_extensions 155
blackfennec.util.parameterized_visitor, 148	blackfennec_doubles.extension_system.double_extensions 155
blackfennec.util.service_locator, 148	blackfennec_doubles.extension_system.double_extensions 155
blackfennec.util.type_comparator, 148	blackfennec_doubles.extension_system.double_extensions 155
blackfennec_doubles, 170	blackfennec_doubles.extension_system.double_extensions 155
blackfennec_doubles.action_system, 150	blackfennec_doubles.extension_system.double_extensions 155
blackfennec_doubles.action_system.double_action, 149	blackfennec_doubles.extension_system.double_extensions 155
blackfennec_doubles.action_system.double_action_registry, 149	blackfennec_doubles.extension_system.double_extensions 155
blackfennec_doubles.action_system.double_context, 149	blackfennec_doubles.extension_system.double_extensions 155
blackfennec_doubles.action_system.double_ui_context, 150	blackfennec_doubles.extension_system.double_extensions 155
blackfennec_doubles.document_system, 154	blackfennec_doubles.extension_system.double_extensions 155
blackfennec_doubles.document_system.double_document	blackfennec_doubles.extension_system.extension_api_fac 155

157
 blackfennec_doubles.layers, 158
 blackfennec_doubles.layers.double_layer,
 158
 blackfennec_doubles.layers.encapsulation_base,
 158
 blackfennec_doubles.layers.encapsulation_base.double_factory_base_visitor,
 157
 blackfennec_doubles.layers.observable,
 158
 blackfennec_doubles.layers.observable.double_observable,
 158
 blackfennec_doubles.presentation_system,
 163
 blackfennec_doubles.presentation_system.double_history_service,
 161
 blackfennec_doubles.presentation_system.double_presenter_factory,
 161
 blackfennec_doubles.presentation_system.double_presenter_registry,
 161
 blackfennec_doubles.presentation_system.double_presenter_view,
 161
 blackfennec_doubles.presentation_system.double_presenter_view_factory,
 162
 blackfennec_doubles.presentation_system.double_presenter_view_factory.type_system.double_type,
 162
 blackfennec_doubles.presentation_system.double_presenter_view_factory.type_system.double_type_loader,
 162
 blackfennec_doubles.presentation_system.double_presenter_view_factory.type_system.double_type_parser,
 162
 blackfennec_doubles.presentation_system.double_presenter_view_factory.type_system.double_type_registry,
 162
 blackfennec_doubles.presentation_system.double_presenter_view_factory.type_system.interpretation,
 162
 blackfennec_doubles.presentation_system.double_presenter_view_factory.type_system.interpretation.double_
 162
 blackfennec_doubles.presentation_system.main_window,
 159
 blackfennec_doubles.presentation_system.main_window.double_document_tabs,
 159
 blackfennec_doubles.presentation_system.navigation_service,
 160
 blackfennec_doubles.presentation_system.navigation_service.double_navigation_service,
 159
 blackfennec_doubles.presentation_system.ui_service,
 161
 blackfennec_doubles.presentation_system.ui_service.double_black_fennec_doubles,
 160
 blackfennec_doubles.presentation_system.ui_service.double_message_overlay,
 160
 blackfennec_doubles.presentation_system.ui_service.double_ui_service,
 160
 blackfennec_doubles.presentation_system.ui_service.double_ui_service_registry,
 160
 blackfennec_doubles.structure, 166
 blackfennec_doubles.structure.double_boolean, 163
 blackfennec_doubles.structure.double_list,
 163
 blackfennec_doubles.structure.double_map,
 164
 blackfennec_doubles.structure.double_null,
 164
 blackfennec_doubles.structure.double_number,
 164
 blackfennec_doubles.structure.double_reference,
 164
 blackfennec_doubles.structure.double_root,
 165
 blackfennec_doubles.structure.double_string,
 165
 blackfennec_doubles.structure.double_structure,
 165
 blackfennec_doubles.structure.reference_navigation,
 165
 blackfennec_doubles.structure.reference_navigation.double_
 165
 blackfennec_doubles.type_system, 169
 blackfennec_doubles.type_system.double_type,
 167
 blackfennec_doubles.type_system.double_type_loader,
 168
 blackfennec_doubles.type_system.double_type_parser,
 168
 blackfennec_doubles.type_system.double_type_registry,
 168
 blackfennec_doubles.type_system.interpretation,
 167
 blackfennec_doubles.type_system.interpretation.double_
 166
 blackfennec_doubles.type_system.interpretation.double_
 166
 blackfennec_doubles.type_system.interpretation.double_
 166
 blackfennec_doubles.type_system.interpretation.double_
 167
 blackfennec_doubles.type_system.interpretation.double_
 167
 blackfennec_doubles.util, 170
 blackfennec_doubles.util.double_change_notification,
 170
 blackfennec_doubles.util.double_message_overlay,
 170
 blackfennec_doubles.util.double_comparable,
 170
 blackfennec_doubles.util.double_meta_info,
 170
 blackfennec_doubles.util.double_observable,
 170
 blackfennec_doubles.util.double_service_locator,
 170

170 tests.blackfennec.extension_system.test_presenter_registry, 180
 tests, 214 tests.blackfennec.extension_system.test_view_factory, 180
 tests.blackfennec, 212 tests.blackfennec.extension_system.test_view_factory_registry, 180
 tests.blackfennec.action_system, 171 tests.blackfennec.layers, 191
 tests.blackfennec.action_system.test_action, 170 tests.blackfennec.layers.encapsulation_base, 185
 tests.blackfennec.action_system.test_action_registry, 171 tests.blackfennec.layers.encapsulation_base.test_base_serializer, 181
 tests.blackfennec.action_system.test_context, 171 tests.blackfennec.layers.encapsulation_base.test_encapsulation_base, 182
 tests.blackfennec.document_system, 177 tests.blackfennec.layers.encapsulation_base.test_list_serializer, 183
 tests.blackfennec.document_system.mime_type, 174 tests.blackfennec.layers.encapsulation_base.test_map_serializer, 184
 tests.blackfennec.document_system.mime_type.json, 174 tests.blackfennec.layers.encapsulation_base.test_reference_serializer, 185
 tests.blackfennec.document_system.mime_type.json.test_json_mime_type, 172 tests.blackfennec.layers.merge, 188
 tests.blackfennec.document_system.mime_type.json.test_json_mime_type_serializer, 172 tests.blackfennec.layers.merge.test_integration_merged_layer, 185
 tests.blackfennec.document_system.mime_type.json.test_json_mime_type_serializer_registry, 173 tests.blackfennec.layers.merge.test_merged_layer, 185
 tests.blackfennec.document_system.mime_type.test_mime_type, 173 tests.blackfennec.layers.merge.test_merged_list, 185
 tests.blackfennec.document_system.mime_type.test_mime_type_registry, 174 tests.blackfennec.layers.merge.test_merged_map, 187
 tests.blackfennec.document_system.resource_type, 176 tests.blackfennec.layers.merge.test_merged_null, 187
 tests.blackfennec.document_system.resource_type.protocols, 175 tests.blackfennec.layers.merge.test_merged_phantom, 187
 tests.blackfennec.document_system.resource_type.protocols.test_file_resource_type, 174 tests.blackfennec.layers.merge.test_merged_structure, 187
 tests.blackfennec.document_system.resource_type.protocols.test_integration_https_resource_type, 175 tests.blackfennec.layers.observable, 190
 tests.blackfennec.document_system.resource_type.protocols.test_integration_https_resource_type_registry, 175 tests.blackfennec.layers.observable.test_integration_https_resource_type_registry, 188
 tests.blackfennec.document_system.resource_type.protocols.test_observable, 175 tests.blackfennec.layers.observable.test_observable, 189
 tests.blackfennec.document_system.test_document_factory, 176 tests.blackfennec.layers.observable.test_observable_base, 189
 tests.blackfennec.document_system.test_document_factory_registry, 176 tests.blackfennec.layers.observable.test_observable_enricher, 190
 tests.blackfennec.document_system.test_document_factory_registry_registry, 177 tests.blackfennec.layers.observable.test_observable_factory, 190
 tests.blackfennec.extension_system, 181 tests.blackfennec.layers.overlay, 191
 tests.blackfennec.extension_system.test_extensions, 177 tests.blackfennec.layers.overlay.test_integration_overlay, 190
 tests.blackfennec.extension_system.test_extensions_registry, 177 tests.blackfennec.layers.overlay.test_overlay, 190
 tests.blackfennec.extension_system.test_extensions_registry_registry, 179 tests.blackfennec.layers.overlay.test_overlay_base, 190
 tests.blackfennec.extension_system.test_extensions_registry_registry_registry, 179 tests.blackfennec.layers.overlay.test_overlay_factory, 190
 tests.blackfennec.extension_system.test_extensions_registry_registry_registry_registry, 179 tests.blackfennec.layers.overlay.test_overlay_factory_registry, 191

tests.blackfennec.presentation_system,	tests.blackfennec.type_system, 209
196	tests.blackfennec.type_system.interpretation,
tests.blackfennec.presentation_system.main_window, 205	tests.blackfennec.type_system.interpretation.test_cove
194	tests.blackfennec.type_system.interpretation.test_black_fennec_view_model,
tests.blackfennec.presentation_system.main_window, 203	tests.blackfennec.type_system.interpretation.test_inte
191	tests.blackfennec.type_system.interpretation.test_inte
tests.blackfennec.presentation_system.main_window, 203	tests.blackfennec.type_system.interpretation.test_inte
193	tests.blackfennec.type_system.interpretation.test_inte
tests.blackfennec.presentation_system.navigation_service,	tests.blackfennec.type_system.interpretation.test_inte
194	tests.blackfennec.type_system.interpretation.test_inte
tests.blackfennec.presentation_system.navigation_service.test_navigation_proxy,	tests.blackfennec.type_system.interpretation.test_offe
194	tests.blackfennec.type_system.interpretation.test_spec
tests.blackfennec.presentation_system.navigation_service.test_navigation_service,	tests.blackfennec.type_system.interpretation.test_spec
194	tests.blackfennec.type_system.interpretation.test_spec
tests.blackfennec.presentation_system.test_history, 205	tests.blackfennec.type_system.test_boolean_type,
195	tests.blackfennec.type_system.test_integration_type_lo
tests.blackfennec.presentation_system.ui_service, 205	tests.blackfennec.type_system.test_list_type,
195	tests.blackfennec.type_system.test_map_type,
tests.blackfennec.presentation_system.ui_service.test_message,	tests.blackfennec.type_system.test_null_type,
194	tests.blackfennec.type_system.test_number_type,
tests.blackfennec.presentation_system.ui_service.test_ui_service,	tests.blackfennec.type_system.test_string_type,
195	tests.blackfennec.type_system.test_type_base,
tests.blackfennec.structure, 203	tests.blackfennec.type_system.test_type_parser,
tests.blackfennec.structure.reference_navigation,	tests.blackfennec.type_system.test_type_registry,
199	tests.blackfennec.util.test_change_notification,
tests.blackfennec.structure.reference_navigation.test_black_fennec_navigator,	tests.blackfennec.util.test_change_notification_dispat
196	tests.blackfennec.util.test_comparable,
tests.blackfennec.structure.reference_navigation.test_black_fennec_navigator,	tests.blackfennec.util.test_deep_copy,
197	tests.blackfennec.util.test_integration_service_locato
tests.blackfennec.structure.reference_navigation.test_black_fennec_navigator,	tests.blackfennec.util.test_integration_service_locato
197	tests.blackfennec.util.test_integration_service_locato
tests.blackfennec.structure.reference_navigation.test_black_fennec_navigator,	tests.blackfennec.util.test_meta_info,
198	tests.blackfennec.util.test_observable,
tests.blackfennec.structure.reference_navigation.test_black_fennec_navigator,	tests.blackfennec.util.test_observable,
198	tests.blackfennec.util.test_observable,
tests.blackfennec.structure.reference_navigation.test_black_fennec_navigator,	tests.blackfennec.util.test_offset_navigation,
198	tests.blackfennec.util.test_offset_navigation,
tests.blackfennec.structure.reference_navigation.test_black_fennec_navigator,	tests.blackfennec.util.test_offset_navigation,
199	tests.blackfennec.util.test_offset_navigation,
tests.blackfennec.structure.test_boolean,	tests.blackfennec.util.test_offset_navigation,
199	tests.blackfennec.util.test_offset_navigation,
tests.blackfennec.structure.test_list,	tests.blackfennec.util.test_offset_navigation,
200	tests.blackfennec.util.test_offset_navigation,
tests.blackfennec.structure.test_map, 200	tests.blackfennec.util.test_offset_navigation,
tests.blackfennec.structure.test_null,	tests.blackfennec.util.test_offset_navigation,
201	tests.blackfennec.util.test_offset_navigation,
tests.blackfennec.structure.test_number,	tests.blackfennec.util.test_offset_navigation,
201	tests.blackfennec.util.test_offset_navigation,
tests.blackfennec.structure.test_reference,	tests.blackfennec.util.test_offset_navigation,
202	tests.blackfennec.util.test_offset_navigation,
tests.blackfennec.structure.test_string,	tests.blackfennec.util.test_offset_navigation,
202	tests.blackfennec.util.test_offset_navigation,
tests.blackfennec.structure.test_structure,	tests.blackfennec.util.test_offset_navigation,
202	tests.blackfennec.util.test_offset_navigation,

tests.test_utils.observer, 214
tests.test_utils.parameterize, 214
tests.test_utils.what_the_farmer_does_not_eat_visitor_factory, 214

N

name (blackfennec.action_system.action.Action property), 104
name (blackfennec.extension_system.extension.Extension property), 115
name (blackfennec.type_system.type.Type property), 143
name (tests.blackfennec.action_system.test_action.ConcreteAction property), 171

navigate() (blackfennec.presentation_system.navigation_service.navigation_proxy.NavigationProxy method), 126
navigate() (blackfennec.presentation_system.navigation_service.navigation_service.NavigationService method), 127
navigate() (blackfennec.structure.reference_navigation.child_navigator.ChildNavigator method), 131
navigate() (blackfennec.structure.reference_navigation.index_of_navigator.IndexOfNavigator method), 131
navigate() (blackfennec.structure.reference_navigation.navigator.Navigator method), 132
navigate() (blackfennec.structure.reference_navigation.parent_navigator.ParentNavigator method), 132
navigate() (blackfennec.structure.reference_navigation.root_navigator.RootNavigator method), 132
navigate() (blackfennec.structure.reference_navigation.sibling_offset_navigator.SiblingOffsetNavigator method), 132
navigate() (blackfennec.structure.reference_navigation.uri_navigator.UriNavigator method), 133
navigate() (blackfennec.type_system.interpretation.interpretation.Interpretation method), 138
navigate() (blackfennec_doubles.presentation_system.navigation_service.NavigationService method), 159
navigate() (blackfennec_doubles.structure.reference_navigation.double_navigator.DoubleNavigator method), 163
navigate() (blackfennec_doubles.type_system.interpretation.double_interpretation.DoubleInterpretation method), 166
navigate() (tests.blackfennec.structure.reference_navigation.number_navigator.NumberNavigator method), 197

navigation_service() (in module tests.blackfennec.presentation_system.main_window.test_document, 103)
NavigationProxy (class in blackfennec.presentation_system.navigation_service.navigation_proxy), 126
NavigationProxyTestSuite (class in tests.blackfennec.presentation_system.navigation_service.test_navigation_proxy, 194)
NavigationService (class in blackfennec.presentation_system.navigation_service.navigation_service), 127
NavigationServiceMock (class in blackfennec_doubles.presentation_system.navigation_service.double_navigation_service, 159)
NavigationServiceTestSuite (class in tests.blackfennec.presentation_system.navigation_service.test_navigation_service, 194)
Navigator (class in blackfennec.structure.reference_navigation.navigator), 132
navigator() (in module tests.blackfennec.structure.reference_navigation.test_navigator, 107)
navigator() (in module tests.blackfennec.structure.test_reference, 202)
NavigatorMock (class in blackfennec_doubles.structure.reference_navigation.double_navigation, 163)
NavigatorSubclass (class in tests.blackfennec.structure.reference_navigation.test_navigator, 197)
new_value (blackfennec.util.change_notification.ChangeNotification property), 146
NOT_COVERED (blackfennec.type_system.interpretation.coverage.Coverage attribute), 138
NonFinalizingStructureMock (class in blackfennec_doubles.structure.double_structure), 165
Null (class in blackfennec.structure.null), 135
NullComparator (class in tests.test_utils.deep_compare), 213
NullMerger (class in blackfennec_doubles.structure.double_structure, 122)
NullMock (class in blackfennec_doubles.structure.double_null), 164
NullTestSuite (class in tests.blackfennec.structure.test_null), 201
NullType (class in blackfennec.type_system.null_type), 142
Number (class in blackfennec.structure.number), 135
NumberComparator (class in tests.blackfennec.structure.test_number, 197)

tests.test_utils.deep_compare), 213
NumberMerger (class in *blackfennec.layers.merge.merger*), 122
NumberMock (class in *blackfennec_doubles.structure.double_number*), 164
NumberTestSuite (class in *tests.blackfennec.structure.test_number*), 201
NumberType (class in *blackfennec.type_system.number_type*), 142

O

Observable (class in *blackfennec.util.observable*), 148
observable_layer() (in module *tests.blackfennec.layers.observable.test_integration_observable_layer*), 188
ObservableBase (class in *blackfennec.layers.observable.observable_base*), 122
ObservableFactoryVisitor (class in *blackfennec.layers.observable.observable_factory_visitor*), 123
ObservableLayer (class in *blackfennec.layers.observable.observable*), 122
ObservableLayerMock (class in *blackfennec_doubles.layers.observable.double_observable*), 158
ObservableMock (class in *blackfennec_doubles.util.double_observable*), 170
ObservableTestSuite (class in *tests.blackfennec.util.test_observable*), 211
observe() (*blackfennec.presentation_system.history_service.HistoryService* method), 129
observe() (*blackfennec_doubles.presentation_system.double_history_service.DoubleHistoryServiceMock* method), 161
Observer (class in *tests.test_utils.observer*), 214
Offer (class in *blackfennec.type_system.interpretation.offer*), 139
offer() (in module *tests.blackfennec.type_system.interpretation.test_offer*), 204
OfferFake (class in *blackfennec_doubles.type_system.interpretation.double_offer*), 167
old_value (*blackfennec.util.change_notification.ChangeNotification* class property), 146
on_changed() (*blackfennec.layers.observable.observable.ObservableLayer* method), 122
on_changed() (*blackfennec_doubles.layers.observable.double_observable.ObservableLayerMock* method), 158
open() (*blackfennec.presentation_system.main_window.black_fennec_view_model.BlackFennecViewModel* method), 125
open_file() (*blackfennec.presentation_system.main_window.black_fennec_view_model.BlackFennecViewModel* method), 125
Overlay (class in *blackfennec.layers.overlay.overlay*), 123
overlay() (in module *tests.blackfennec.layers.overlay.test_overlay*), 190
overlay_base() (in module *tests.blackfennec.layers.overlay.test_overlay_base*), 190
OverlayBase (class in *blackfennec.layers.overlay.overlay_base*), 123
OverlayFactoryVisitor (class in *blackfennec.layers.overlay.overlay_factory_visitor*), 123
OverlayFactoryVisitorTestSuite (class in *tests.blackfennec.layers.overlay.test_overlay_factory_visitor*), 191

P

ParameterizedVisitor (class in *blackfennec.util.parameterized_visitor*), 148
parent (*blackfennec.layers.encapsulation_base.encapsulation_base.EncapsulationBase* property), 117
parent (*blackfennec.layers.merge.merged_phantom.MergedPhantom* property), 120
parent (*blackfennec.layers.merge.merged_structure.MergedStructure* property), 120
parent (*blackfennec.structure.structure.Structure* property), 136
parent (*blackfennec.type_system.type.Type* property), 143
parent (*blackfennec_doubles.structure.double_structure.StructureMock* property), 165
parent() (in module *tests.blackfennec.structure.reference_navigation.test_parent_navigator*), 198
parent_navigator() (in module *tests.blackfennec.structure.reference_navigation.test_parent_navigator*), 198
ParentNavigator (class in *blackfennec.structure.reference_navigation.parent_navigator*), 132
parse() (*blackfennec.type_system.type_parser.TypeParser* method), 145
parser() (in module *tests.blackfennec.layers.merge.test_merged_layer*), 186
pattern (*blackfennec.type_system.string_type.StringType* property), 143
presenter_registry (*blackfennec.presentation_system.extension_api.ExtensionApi* property), 116
presenter_registry() (*blackfennec.presentation_system.extension_api.ExtensionApi* method), 116
presenter_registry() (*blackfennec.presentation_system.test_extension_api.TestExtensionApi* method), 116

178
 presenter_registry() (in module REDO (tests.blackfennec.layers.observable.test_integration_observable_layer), tests.blackfennec.presentation_system.main_window.test_blackfennec_view_model), 188

191
 presenter_registry() (in module REDO (tests.blackfennec.presentation_system.test_history_service.Action attribute), 195
 tests.blackfennec.presentation_system.main_window.test_blackfennec_view_model.presentation_system.history_service.HistoryService method), 129

193
 PresenterFactoryMock (class in blackfennec_doubles.presentation_system.double_presenter_factory) redo() (blackfennec.presentation_system.main_window.black_fennec_view_model), 125

161
 PresenterRegistry (class in blackfennec.presentation_system.presenter_registry), redo() (blackfennec_doubles.presentation_system.double_history_service.method), 161

129
 PresenterRegistryMock (class in blackfennec_doubles.presentation_system.double_presenter_registry), Reference (class in blackfennec.structure.reference), 135

161
 PresenterRegistryTestSuite (class in tests.blackfennec.extension_system.test_presenter_registry), Reference (class in blackfennec.structure.reference), 202

180
 presenters (blackfennec.presentation_system.presenter_registry.PresenterRegistry property), ReferenceKey (blackfennec_doubles.document_system.mime_type.json.double_json_reference attribute), 151

presenters (blackfennec_doubles.presentation_system.double_presenter_registry.PresenterRegistry property), ReferenceComparator (class in blackfennec.layers.encapsulation_base.reference_encapsulation_base), 213

PresenterViewMock (class in blackfennec_doubles.presentation_system.double_presenter_view), ReferenceEncapsulationBase (class in blackfennec.layers.encapsulation_base.reference_encapsulation_base), 119

161
 PresenterViewModelMock (class in blackfennec_doubles.presentation_system.double_presenter_view_model), ReferenceInstanceMock (class in blackfennec_doubles.structure.double_reference), 164

properties (blackfennec.type_system.map_type.MapType property), ReferenceMock (class in blackfennec_doubles.structure.double_reference), 164

142
 protocols (blackfennec.document_system.resource_type.protocols.binary_resource_type.BinaryResourceType), ReferenceType (class in blackfennec.type_system.type_reference_type.ReferenceType), 142

protocols (blackfennec.document_system.resource_type.protocols.file_resource_type.FileResourceType), refresh_extensions() (blackfennec_doubles.extension_source.Extension method), 157

protocols (blackfennec.document_system.resource_type.protocols.http_resource_type.HttpResourceType), register() (blackfennec.extension_system.extension_registry.ExtensionRegistry method), 157

protocols (blackfennec.document_system.resource_type.resource_type.ResourceType), register() (blackfennec_doubles.extension_registry.ExtensionRegistry method), 157

protocols (blackfennec_doubles.document_system.resource_type.double_double_extension_registry.ExtensionRegistry method), 157

160
 random_string() (in module tests.blackfennec.layers.observable.test_integration_observable_layer), register_action() (blackfennec.action_system.action_registry.ActionRegistry method), 104

188
 random_string() (in module tests.blackfennec.presentation_system.test_history_service), register_action() (blackfennec_doubles.action_system.double_action_registry.ActionRegistry method), 149

196
 raw_json() (in module tests.blackfennec.document_system.mime_type.json.test_json_mime_type), register_actions(), (blackfennec_doubles.action_system.double_action_registry.ActionRegistry method), 149

<i>nec.extension_system.extension.Extension</i> method), 115	<i>tests.blackfennec.extension_system.test_extension_service</i> , 179
register_document() <i>nec.document_system.document_registry.DocumentRegistry</i> method), 114	(blackfen- registry() (in module <i>tests.blackfennec.extension_system.test_view_factory_registry</i>), 180
register_document() <i>nec_doubles.document_system.double_document_registry.DocumentRegistryMock</i> method), 154	(blackfen- RELATIVE_POINTER_PATTERN (blackfen- attribute), 106
register_extension_source() <i>nec_doubles.extension_system.double_extension_source_registry.ExtensionSystemRegistryMock</i> method), 157	(blackfen- RELATIVE_POINTER_PATTERN (blackfen- attribute), 107
register_message_overlay() <i>nec.presentation_system.ui_service.ui_service.UiService</i> method), 128	(blackfen- REMOVE_ITEM (<i>tests.blackfennec.layers.observable.test_integration_observa</i> attribute), 188
register_mime_type() <i>nec.document_system.mime_type.mime_type_registry.MimeTypeRegistry</i> method), 109	remove_item() (blackfen- <i>nec.layers.encapsulation_base.list_encapsulation_base.ListEncap</i> TypeRegistry), 118
register_mime_type() <i>nec_doubles.document_system.mime_type.double_mime_type_registry.MimeTypeRegistryMock</i> method), 152	remove_item() (blackfen- <i>nec.layers.encapsulation_base.map_encapsulation_base.MapEncap</i> MimeTypeRegistryMock
register_presenter() <i>nec.presentation_system.presenter_registry.PresenterRegistry</i> method), 129	remove_item() (blackfennec.structure.map.Map method), 134
register_presenter() <i>nec_doubles.presentation_system.double_presenter_registry.PresenterRegistryMock</i> method), 161	remove_item() (blackfen- <i>nec.layers.encapsulation_base.map_encapsulation_base.MapEncap</i> method), 163
register_presenters() <i>nec.extension_system.extension.Extension</i> method), 115	remove_item() (blackfen- <i>nec_doubles.structure.double_map.MapMock</i> method), 164
register_resource_type() <i>nec.document_system.resource_type.resource_type_registry.ResourceTypeRegistry</i> method), 113	rename_key() (blackfen- <i>nec.layers.encapsulation_base.map_encapsulation_base.MapEncap</i> method), 119
register_resource_type() <i>nec_doubles.document_system.resource_type.double_resource_type_registry.ResourceTypeRegistryMock</i> method), 153	rename_key() (blackfennec.structure.map.Map method), 119
register_type() <i>nec.type_system.type_registry.TypeRegistry</i> method), 145	replace_item() (blackfen- <i>nec.layers.encapsulation_base.list_encapsulation_base.ListEncap</i> method), 118
register_type() <i>nec_doubles.type_system.double_type_registry.TypeRegistryMock</i> method), 168	replace_item() (blackfen- <i>nec.layers.encapsulation_base.map_encapsulation_base.MapEncap</i> method), 119
register_type_view_factory() <i>nec.presentation_system.type_view_factory_registry.TypeViewFactoryRegistry</i> method), 131	replace_item() (blackfennec.structure.list.List method), 134
register_type_view_factory() <i>nec_doubles.presentation_system.double_view_factory_registry.TypeViewFactoryRegistryMock</i> method), 162	replace_item() (blackfen- <i>nec.layers.encapsulation_base.map_encapsulation_base.MapEncap</i> method), 134
register_types() <i>nec.extension_system.extension.Extension</i> method), 115	required_elements (blackfen- <i>nec.type_system.map_type.MapType</i> property), 141
register_types() <i>nec.extension_system.extension.Extension</i> method), 115	required_properties (blackfen- <i>nec.type_system.map_type.MapType</i> property), 142
register_view_factories() <i>nec.extension_system.extension.Extension</i> method), 115	resolve() (<i>blackfennec.layers.encapsulation_base.reference_encapsulatio</i> method), 119
registry() (in module	resolve() (<i>blackfennec.structure.reference.Reference</i> method), 135

resolve() (blackfennec_doubles.structure.double_reference_root() (in module tests.blackfennec.presentation_system.ui_service.test_ui_service), 164
 resource_type (blackfennec.document_system.document.Document property), 113
 resource_type() (tests.blackfennec.document_system.test_document_factory.TestDocument method), 176
 resource_type_registry (blackfennec.extension_system.extension_api.ExtensionApi property), 116
 resource_type_registry() (in module tests.blackfennec.document_system.test_document_factory), 176
 resource_type_registry() (in module tests.blackfennec.extension_system.test_extension_registry.TestExtensionRegistry), 178
 resource_types (blackfennec.document_system.resource_type.resource_type_registry.ResourceTypeRegistry property), 113
 resource_types (blackfennec_doubles.document_system.resource_type.double_resource_type_registry.ResourceTypeRegistryMock property), 153
 ResourceType (class in blackfennec.document_system.resource_type), 112
 ResourceTypeMock (class in blackfennec_doubles.document_system.resource_type.double_resource_type_registry), 153
 ResourceTypeRegistry (class in blackfennec.document_system.resource_type), 112
 ResourceTypeRegistryMock (class in blackfennec_doubles.document_system.resource_type.double_resource_type_registry), 153
 ResourceTypeRegistryTestSuite (class in tests.blackfennec.document_system.resource_type.test_resource_type_registry), 175
 ResourceTypeTestSuite (class in tests.blackfennec.document_system.resource_type.test_resource_type_registry), 175
 RESULT (tests.blackfennec.layers.observable.test_integration_observable_layer.Action attribute), 188
 RESULT (tests.blackfennec.presentation_system.test_history_save_document_as() attribute), 196
 root (blackfennec.layers.encapsulation_base.encapsulation_base.EncapsulationBase property), 117
 root (blackfennec.layers.merge.merged_structure.MergedStructure property), 120
 root (blackfennec.structure.structure.Structure property), 136
 root (blackfennec_doubles.structure.double_root.RootMock property), 165
 root (blackfennec_doubles.structure.double_structure.StructureMock property), 165
 root() (in module tests.blackfennec.structure.reference_navigation.test_root_navigation), 198
 root() (in module tests.blackfennec.structure.reference_navigation.test_root_navigation), 198
 root_child() (in module tests.blackfennec.structure.reference_navigation.test_root_navigation), 198
 root_navigator() (in module tests.blackfennec.structure.reference_navigation.test_root_navigation), 198
 RootMock (class in blackfennec_doubles.structure.double_root), 165
 RootNavigator (class in blackfennec.structure.reference_navigation.root_navigator), 132
 satisfies() (blackfennec.presentation_system.resource_type_registry.ResourceTypeRegistryMock method), 130
 satisfies() (blackfennec_doubles.type_system.interpretation.double_offer.OfferFake method), 167
 save() (blackfennec.document_system.document.Document method), 113
 save() (blackfennec.presentation_system.main_window.black_fennec_view_model.BlackFennecViewMethod), 125
 save() (blackfennec_doubles.document_system.double_document.Document method), 154
 save_all() (blackfennec.presentation_system.main_window.black_fennec_view_model.BlackFennecViewMethod), 125
 save_as() (blackfennec.presentation_system.main_window.black_fennec_view_model.BlackFennecViewMethod), 125
 save_document() (blackfennec.presentation_system.main_window.document_tab.DocumentTabMethod), 125
 save_document() (blackfennec.presentation_system.main_window.double_document.DoubleDocumentMethod), 159
 save_document_as() (blackfennec.presentation_system.main_window.document_tab.DocumentTabMethod), 125
 save_document_as() (blackfennec_doubles.presentation_system.main_window.double_document.DoubleDocumentMethod), 159
 serialize() (blackfennec.document_system.mime_type.json.json_pointer_serializer.JsonPointerSerializer class method), 107
 serialize() (blackfennec.document_system.mime_type.json.json_reference_serializer.JsonReferenceSerializer class method), 108

serialize()	(blackfen-	142	
<i>nec.structure.structure_serializer.StructureSerializer</i>	set_structure()		(blackfen-
method), 137			<i>nec_doubles.presentation_system.double_presenter_view_model.</i>
serialize()	(blackfen-	method), 162	
<i>nec_doubles.document_system.mime_type.json.double_json_double_serializer_mock</i>	setUp()	(tests.blackfennec.double_presenter_view_model.test_mime_type.M	
method), 151		method), 174	
serialize()	(blackfen-	setUp()	(tests.blackfennec.document_system.resource_type.protocols.test_
<i>nec_doubles.document_system.mime_type.json.double_json_double_serializer.JsonReferenceSerializerMock</i>	setUp()		
method), 151		setUp()	(tests.blackfennec.document_system.resource_type.protocols.test_
serialize()	(blackfen-	method), 175	
<i>nec_doubles.document_system.mime_type.json.double_json_double_serializer.JsonReferenceSerializerMock</i>	setUp()	(tests.blackfennec.layers.serialization_mock_base.test_base_factory_vi	
method), 151		method), 181	
service()	(in	module	setUp()
<i>tests.blackfennec.extension_system.test_extension_service</i>),			(tests.blackfennec.layers.overlay.test_overlay_factory_visitor.Over
179			setUp()
service_locator()	(in	module	method), 199
<i>tests.blackfennec.presentation_system.main_window_controller.test_null.NullTestSuite</i>			
191			setUp()
ServiceLocator	(class	in	blackfen-
<i>nec.util.service_locator</i>), 148			setUp()
ServiceLocatorMock	(class	in	blackfen-
<i>nec_doubles.util.double_service_locator</i>),			setUp()
170			method), 202
services	(<i>blackfennec_doubles.presentation_system.ui_service.double_ui_service_registry.UiServiceRegistryMock</i>		
property), 160			setUp()
set_clipboard()	(blackfen-	method), 207	
<i>nec.presentation_system.ui_service.ui_service.UiService</i>	setUp()	(tests.blackfennec.type_system.test_list_type.ListTypeTestSuite	
class method), 128		method), 208	
set_clipboard()	(blackfen-	setUp()	(tests.blackfennec.util.test_deep_copy.TestDeepCopyVisitorTestSu
<i>nec_doubles.presentation_system.ui_service.double_ui_service.UiServiceMock</i>			
method), 160			show()
set_is_child_optional()	(blackfen-	method), 162	
<i>nec.type_system.list_type.ListType</i>	method),	show()	(blackfennec_doubles.presentation_system.double_structure_presen
141			method), 162
set_is_child_optional()	(blackfen-	show_message()	(blackfen-
<i>nec.type_system.map_type.MapType</i>	method),		<i>nec.presentation_system.ui_service.ui_service.UiService</i>
142			method), 128
set_navigation_service()	(blackfen-	show_message()	(blackfen-
<i>nec.type_system.interpretation.interpretation.Interpretation</i>			<i>nec_doubles.presentation_system.ui_service.double_ui_service.U</i>
method), 138			method), 160
set_navigation_service()	(blackfen-	show_mime_type_selection_dialog()	(blackfen-
<i>nec_doubles.type_system.interpretation.double_interpretation_mock</i>			<i>nec_doubles.presentation_system.ui_service.ui_service.UiService</i>
method), 166			method), 128
set_presenter()	(blackfen-	SiblingOffsetNavigator	(class
<i>nec.presentation_system.navigation_service.navigation_service.NavigationService</i>			in
method), 127			<i>blackfen-</i>
set_presenter()	(blackfen-	specification	(blackfen-
<i>nec_doubles.presentation_system.navigation_service.double_navigation_service.NavigationServiceMock</i>			<i>Interpretation</i>
method), 159			property), 138
set_required()	(blackfen-	Specification	(class
<i>nec.type_system.list_type.ListType</i>	method),		in
141			<i>blackfen-</i>
set_required()	(blackfen-	SpecificationMock	(class
<i>nec.type_system.map_type.MapType</i>	method),		in
			<i>blackfen-</i>
			<i>nec_doubles.type_system.interpretation.double_specification</i>),

167	SpecificationTestSuite	(class in module tests.blackfennec.type_system.interpretation.test_specification), 167	199	structure()	(in module tests.blackfennec.structure.test_reference), 199
205	specificity	(blackfennec.type_system.interpretation.offer.Offer property), 140	202	structure_serializer()	(in module tests.blackfennec.document_system.mime_type.json.test_json_minimal), 202
	state	(blackfennec.extension_system.extension.Extension property), 115		structure_serializer()	(in module tests.blackfennec.document_system.mime_type.json.test_structure), 173
	String	(class in blackfennec.structure.string), 136		StructureComparator	(class in module tests.test_utils.deep_compare), 213
	StringComparator	(class in module tests.test_utils.deep_compare), 213		StructureInstanceMock	(class in blackfennec_doubles.structure.double_structure), 165
	StringMerger	(class in blackfennec.layers.merge.merger), 122		StructureMock	(class in blackfennec_doubles.structure.double_structure), 165
	StringMock	(class in blackfennec_doubles.structure.double_string), 165		StructurePresenterMock	(class in blackfennec_doubles.presentation_system.double_structure_presenter), 162
	StringTestSuite	(class in module tests.blackfennec.structure.test_string), 202		StructureSerializer	(class in blackfennec.structure.structure_serializer), 136
	StringType	(class in blackfennec.type_system.string_type), 143		StructureSerializerMock	(class in blackfennec_doubles.document_system.mime_type.json.double_structure_serializer), 162
	StringTypeTestSuite	(class in module tests.blackfennec.type_system.test_string_type), 208		StructureTestMixin	(class in module tests.blackfennec.structure.test_structure), 202
	structure	(blackfennec.action_system.context.Context attribute), 104		StructureViewFactory	(class in blackfennec.presentation_system.structure_view_factory), 129
	structure	(blackfennec.layers.encapsulation_base.encapsulation_base.EncapsulationBase property), 117		StructureViewFactoryMock	(class in blackfennec_doubles.presentation_system.double_structure_view_factory), 162
	structure	(blackfennec.layers.merge.merged_phantom.MergedPhantom property), 120		Subject	(blackfennec.layers.encapsulation_base.encapsulation_base.EncapsulationBase property), 117
	structure	(blackfennec.layers.merge.merged_structure.MergedStructure property), 121		SubjectMock	(blackfennec_doubles.list_encapsulation_base.list_encapsulation_base.ListEncapsulationBase property), 118
	structure	(blackfennec.structure.structure.Structure property), 136		subject	(blackfennec.layers.encapsulation_base.map_encapsulation_base.MapEncapsulationBase property), 119
	structure	(blackfennec.type_system.interpretation.interpretation.Interpretation property), 138		subject	(blackfennec.layers.merge.merged_structure.MergedStructure property), 121
	structure	(blackfennec_doubles.structure.double_structures.DoubleStructures property), 165		subject	(blackfennec.type_system.interpretation.offer.Offer property), 140
	structure	(blackfennec_doubles.type_system.interpretation.offer.Offer property), 166		subject	(blackfennec.type_system.type.Type property), 143
	Structure	(class in blackfennec.structure.structure), 136		subject()	(in module tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base), 183
	structure()	(in module tests.blackfennec.layers.merge.test_merged_layer), 186		subject()	(in module tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base), 184
	structure()	(in module tests.blackfennec.layers.observable.test_integration_observation_layer), 189		subject()	(in module tests.blackfennec.layers.encapsulation_base.test_offset_navigation), 184
	structure()	(in module tests.blackfennec.presentation_system.test_history_navigation), 196		subject()	(in module tests.blackfennec.layers.encapsulation_base.test_offset_navigation), 184
	structure()	(in module tests.blackfennec.structure.reference_navigation.test_navigation), 198		subject()	(in module tests.blackfennec.layers.encapsulation_base.test_offset_navigation), 184
	structure()	(in module tests.blackfennec.structure.reference_navigation.test_navigation), 198		subject()	(in module tests.blackfennec.layers.encapsulation_base.test_offset_navigation), 184

tests.blackfennec.type_system.interpretation.test_offer), 204
 204
 super (*blackfennec.type_system.type.Type* property), 144
T
 tearDown() (*tests.blackfennec.document_system.resource_type.protocol_test_file_resource_type.FileResourceTypeTestSuite* method), 174
 test_accept() (in module *tests.blackfennec.structure.test_list*), 200
 test_accept() (in module *tests.blackfennec.structure.test_map*), 200
 test_accept() (*tests.blackfennec.structure.test_boolean.BooleanTestSuite* method), 199
 test_add_item_does_set_parent() (in module *tests.blackfennec.structure.test_list*), 200
 test_add_item_does_set_parent() (in module *tests.blackfennec.structure.test_map*), 200
 test_add_item_item() (in module *tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base*), 183
 test_add_item_item() (in module *tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base*), 184
 test_add_item_item_already_encapsulated() (in module *tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base*), 183
 test_add_item_item_already_encapsulated() (in module *tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base*), 184
 test_add_item_logs_on_key_occupied() (in module *tests.blackfennec.structure.test_map*), 200
 test_add_item_logs_on_parent_not_none() (in module *tests.blackfennec.structure.test_list*), 200
 test_add_item_raises_assertion_error() (in module *tests.blackfennec.structure.test_map*), 200
 test_add_item_throws_on_key_occupied() (in module *tests.blackfennec.structure.test_map*), 200
 test_add_item_throws_on_parent_not_none() (in module *tests.blackfennec.structure.test_list*), 200
 test_add_item_throws_on_parent_not_none() (in module *tests.blackfennec.structure.test_map*), 200
 test_all_satisfying_offers_are_returned() (in module *tests.blackfennec.type_system.interpretation.test_interpretation_service*), 204
 test_auction() (in module *tests.blackfennec.type_system.interpretation.test_integration_interpretation_service*), 203
 test_auction_with_no_fitting_offers() (in module *tests.blackfennec.type_system.interpretation.test_interpretation_service*), 206
 test_bind_kwargs() (*tests.blackfennec.util.test_observable.ObservableTestSuite* method), 211
 test_calculate_coverage_map_full_coverage() (*tests.blackfennec.type_system.test_map_type.MapTypeTestSuite* method), 207
 test_calculate_coverage_map_half_coverage() (*tests.blackfennec.type_system.test_map_type.MapTypeTestSuite* method), 207
 test_calculate_coverage_map_incompatible() (*tests.blackfennec.type_system.test_map_type.MapTypeTestSuite* method), 207
 test_calculate_coverage_map_third_coverage() (*tests.blackfennec.type_system.test_map_type.MapTypeTestSuite* method), 207
 test_calculate_coverage_wrong_type() (*tests.blackfennec.type_system.test_list_type.ListTypeTestSuite* method), 206
 test_calculate_coverage_wrong_type() (*tests.blackfennec.type_system.test_map_type.MapTypeTestSuite* method), 207
 test_calculate_coverage_wrong_type() (*tests.blackfennec.type_system.test_string_type.StringTypeTestSuite* method), 208
 test_can_accept() (*tests.blackfennec.structure.test_structure.StructureTestSuite* method), 202
 test_can_accept_and_return_self() (in module *tests.blackfennec.layers.overlay.test_overlay_base*), 190
 test_can_accept_correctly() (in module *tests.blackfennec.layers.merge.test_merged_structure*), 187
 test_can_accept_visitor() (in module *tests.blackfennec.layers.merge.test_merged_null*), 187
 test_can_access_parent() (in module *tests.blackfennec.layers.merge.test_merged_layer*), 186
 test_can_access_super() (in module *tests.blackfennec.layers.merge.test_merged_layer*), 186
 test_can_access_super_super() (in module *tests.blackfennec.layers.merge.test_merged_layer*), 186
 test_can_activate() (in module *tests.blackfennec.extension_system.test_extension*), 177
 test_can_add_all_core_types() (in module *tests.blackfennec.structure.test_list*), 200
 test_can_add_all_core_types() (in module *tests.blackfennec.structure.test_map*), 200
 test_can_add_element() (*tests.blackfennec.type_system.test_list_type.ListTypeTestSuite* method), 206

test_can_add_item()	(in module tests.blackfennec.structure.test_map),	200	test_can_construct()	(in module tests.blackfennec.document_system.mime_type.json.test_json_refe)	173
test_can_add_item_item()	(in module tests.blackfennec.structure.test_list),	200	test_can_construct()	(in module tests.blackfennec.document_system.mime_type.json.test_structure)	173
test_can_add_optional_element()	(tests.blackfennec.type_system.test_list_type.ListTypeTestSuite	method), 206	test_can_construct()	(in module tests.blackfennec.document_system.test_document_factory),	176
test_can_add_required_element()	(tests.blackfennec.type_system.test_list_type.ListTypeTestSuite	method), 206	test_can_construct()	(in module tests.blackfennec.extension_system.test_extension_api),	178
test_can_append()	(in module tests.blackfennec.presentation_system.test_history_service),	196	test_can_construct()	(in module tests.blackfennec.extension_system.test_extension_registry),	179
test_can_apply()	(in module tests.blackfennec.layers.observable.test_observable),	189	test_can_construct()	(in module tests.blackfennec.extension_system.test_view_factory),	180
test_can_apply()	(in module tests.blackfennec.layers.overlay.test_overlay),	190	test_can_construct()	(in module tests.blackfennec.layers.encapsulation_base.test_list_encapsulatio	183
test_can_attach_tab()	(in module tests.blackfennec.presentation_system.main_window.test_blackfennec_model)	191	test_can_construct()	(in module tests.blackfennec.layers.encapsulation_base.test_map_encapsulat	184
test_can_be_created()	(tests.blackfennec.type_system.test_number_type.TestNumberType	method), 208	test_can_construct()	(in module tests.blackfennec.layers.merge.test_merged_list),	187
test_can_calculate_coverage()	(tests.blackfennec.type_system.test_string_type.StringTypeTestSuite	method), 208	test_can_construct()	(in module tests.blackfennec.layers.merge.test_merged_map),	187
test_can_calculate_coverage_normal_string()	(tests.blackfennec.type_system.test_string_type.StringTypeTestSuite	method), 208	test_can_construct()	(in module tests.blackfennec.layers.merge.test_merged_null),	187
test_can_calculate_coverage_of_structure()	(in module tests.blackfennec.type_system.test_type_base),	209	test_can_construct()	(in module tests.blackfennec.layers.observable.test_observable),	189
test_can_calculate_coverage_pattern_match()	(tests.blackfennec.type_system.test_string_type.StringTypeTestSuite	method), 208	test_can_construct()	(in module tests.blackfennec.layers.observable.test_observable_base),	189
test_can_calculate_coverage_pattern_mismatch()	(tests.blackfennec.type_system.test_string_type.StringTypeTestSuite	method), 208	test_can_construct()	(in module tests.blackfennec.layers.observable.test_observable_entry),	190
test_can_change_parent()	(tests.blackfennec.structure.test_boolean.BooleanTestSuite	method), 199	test_can_construct()	(in module tests.blackfennec.layers.observable.test_observable_factory_visit	190
test_can_close_file()	(in module tests.blackfennec.presentation_system.main_window.test_blackfennec_model)	191	test_can_construct()	(in module tests.blackfennec.layers.overlay.test_overlay),	190
test_can_compare_merged_objects()	(in module tests.blackfennec.layers.merge.test_merged_layer),	186	test_can_construct()	(in module tests.blackfennec.layers.overlay.test_overlay_base),	190
test_can_construct()	(in module tests.blackfennec.action_system.test_action_registry),	171	test_can_construct()	(in module tests.blackfennec.presentation_system.test_history_service),	190
test_can_construct()	(in module tests.blackfennec.document_system.mime_type.json.test_json_refe	method), 173	test_can_construct()	(in module tests.blackfennec.presentation_system.test_history_service),	190

196 (tests.blackfennec.structure.test_boolean.BooleanTestSuite
test_can_construct() (in module method), 199
tests.blackfennec.presentation_system.ui_service.test_can_construct()
194 (tests.blackfennec.structure.test_null.NullTestSuite
test_can_construct() (in module method), 201
tests.blackfennec.presentation_system.ui_service.test_can_construct()
195 (tests.blackfennec.structure.test_number.NumberTestSuite
test_can_construct() (in module method), 201
tests.blackfennec.structure.reference_navigation.test_can_construct()
196 (tests.blackfennec.structure.test_string.StringTestSuite
test_can_construct() (in module method), 202
tests.blackfennec.structure.reference_navigation.test_can_construct()
197 (tests.blackfennec.type_system.test_boolean_type.TestBooleanType
test_can_construct() (in module method), 205
tests.blackfennec.structure.reference_navigation.test_can_construct()
198 (tests.blackfennec.type_system.test_list_type.ListTypeTestSuite
test_can_construct() (in module method), 206
tests.blackfennec.structure.reference_navigation.test_can_construct()
198 (tests.blackfennec.type_system.test_map_type.MapTypeTestSuite
test_can_construct() (in module method), 207
tests.blackfennec.structure.reference_navigation.test_can_construct()
198 (tests.blackfennec.type_system.test_string_type.StringTypeTestSuite
test_can_construct() (in module method), 208
tests.blackfennec.structure.reference_navigation.test_can_construct()
199 test_can_construct_action() (in module
tests.blackfennec.action_system.test_action),
171
test_can_construct() (in module
tests.blackfennec.structure.test_list), 200 test_can_construct_document_tab() (in module
tests.blackfennec.presentation_system.main_window.test_document
193
test_can_construct() (in module
tests.blackfennec.structure.test_map), 200 test_can_construct_extension() (in module
tests.blackfennec.extension_system.test_extension),
177
test_can_construct() (in module
tests.blackfennec.type_system.test_null_type),
207 test_can_construct_from_dict() (in module
tests.blackfennec.structure.test_map), 200
test_can_construct() (in module
tests.blackfennec.type_system.test_type_base),
209 test_can_construct_from_list() (in module
tests.blackfennec.structure.test_list), 200
test_can_construct() (in module
tests.blackfennec.util.test_change_notification),
210 test_can_construct_not_covered()
(tests.blackfennec.type_system.interpretation.test_coverage.Coverage
method), 203
test_can_construct() (in module
tests.blackfennec.util.test_change_notification_dispatcher),
210 test_can_construct_service_locator() (in mod-
ule tests.blackfennec.util.test_integration_service_locator),
211
test_can_construct() (in module
tests.blackfennec.util.test_change_notification_dispatcher),
210 test_can_copy_view_model() (in module
tests.blackfennec.presentation_system.main_window.test_black_fe
191
test_can_construct() (in module
tests.blackfennec.document_system.test_document),
176 test_can_create_visitor() (in module
tests.blackfennec.type_system.test_string_type.StringTypeTestSuite
method), 208
test_can_construct() (in module
tests.blackfennec.layers.encapsulation_base.test_base_creator_base.CreatorBase
method), 181 test_can_create_visitor() (in module
tests.blackfennec.type_system.test_list_type.ListTypeTestSuite
method), 206
test_can_construct() (in module
tests.blackfennec.layers.overlay.test_overlay_factory),
191 test_can_create_visitor() (in module
tests.blackfennec.type_system.test_map_type.MapTypeTestSuite
method), 207

test_can_create_instance() (in module tests.blackfennec.type_system.test_string_type.StringTypeTestSuite), 208

test_can_create_interpretation() (in module tests.blackfennec.type_system.interpretation.test_interpretation_service_blackfennec), 204

test_can_create_layer() (in module tests.blackfennec.layers.observable.test_integration_observable_blackfennec), 189

test_can_create_offer() (in module tests.blackfennec.type_system.interpretation.test_offer), 204

test_can_create_presenter() (in module tests.blackfennec.presentation_system.main_window.test_document_blackfennec), 193

test_can_create_view() (in module tests.blackfennec.extension_system.test_view_factory), 180

test_can_create_view_factory_registry() (in module tests.blackfennec.extension_system.test_view_factory_registry_blackfennec), 180

test_can_deactivate() (in module tests.blackfennec.extension_system.test_extension), 177

test_can_default_construct() (in module tests.blackfennec.structure.test_boolean.BooleanTestSuite), 199

test_can_default_construct() (in module tests.blackfennec.structure.test_null.NullTestSuite), 201

test_can_default_construct() (in module tests.blackfennec.structure.test_number.NumberTestSuite), 201

test_can_default_construct() (in module tests.blackfennec.structure.test_string.StringTestSuite), 202

test_can_default_create() (in module tests.blackfennec.type_system.interpretation.test_specification_blackfennec), 205

test_can_deregister_action() (in module tests.blackfennec.action_system.test_action_registry), 171

test_can_deregister_message_overlay() (in module tests.blackfennec.presentation_system.ui_service.test_ui_service_blackfennec), 195

test_can_deregister_view_factory() (in module tests.blackfennec.extension_system.test_view_factory_registry_blackfennec), 180

test_can_detach_tab() (in module tests.blackfennec.presentation_system.main_window.test_blackfennec_layer_blackfennec), 191

test_can_execute() (in module tests.blackfennec.action_system.test_action), 171

test_can_get_about_window_view_model() (in module tests.blackfennec.presentation_system.main_window.test_blackfennec), 191

test_can_get_action_name() (in module tests.blackfennec.presentation_system.ui_service.test_message), 194

test_can_get_action_registry() (in module tests.blackfennec.extension_system.test_extension_api), 178

test_can_get_action_target() (in module tests.blackfennec.presentation_system.ui_service.test_message), 194

test_can_get_actions() (in module tests.blackfennec.action_system.test_action_registry), 171

test_can_get_child_of_root() (in module tests.blackfennec.layers.merge.test_integration_merged), 185

test_can_get_dependencies() (in module tests.blackfennec.extension_system.test_extension), 177

test_can_get_document_factory() (in module tests.blackfennec.extension_system.test_extension_api), 178

test_can_get_document_registry() (in module tests.blackfennec.extension_system.test_extension_api), 178

test_can_get_extensions() (in module tests.blackfennec.extension_system.test_extension_registry), 179

test_can_get_factory_if_exists() (in module tests.blackfennec.extension_system.test_view_factory_registry_blackfennec), 180

test_can_get_interpretation_service() (in module tests.blackfennec.extension_system.test_extension_api), 178

test_can_get_is_active() (in module tests.blackfennec.presentation_system.test_extension), 177

test_can_get_merged_children() (in module tests.blackfennec.layers.merge.test_integration_merged), 185

test_can_get_mime_type_registry() (in module tests.blackfennec.extension_system.test_extension_api), 178

test_can_get_name() (in module tests.blackfennec.extension_system.test_extension), 177

test_can_get_parent() (in module tests.blackfennec.layers.merge.test_integration_merged), 185

test_can_get_parent() (in module tests.blackfennec.layers.merge.test_merged_structure), 187

test_can_get_parent_if_no_parent() (in module test_can_get_type_registry() (in module tests.blackfennec.layers.merge.test_integration_merged), 185), tests.blackfennec.extension_system.test_extension_api), 178

test_can_get_presenter_registry() (in module test_can_get_ui_service_registry() (in module tests.blackfennec.extension_system.test_extension_api), 178), tests.blackfennec.extension_system.test_extension_api), 179

test_can_get_repr() (in module test_can_get_value() (in module tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base), 183), tests.blackfennec.layers.merge.test_merged_list), 187

test_can_get_repr() (in module test_can_get_value() (in module tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base), 184), tests.blackfennec.layers.merge.test_merged_map), 187

test_can_get_repr() (in module test_can_get_value() (in module tests.blackfennec.layers.overlay.test_overlay_base), 190), tests.blackfennec.layers.merge.test_merged_null), 187

test_can_get_repr() (in module test_can_get_value() (in module tests.blackfennec.type_system.test_list_type.ListTypeTestSuite method), 206), tests.blackfennec.layers.merge.test_merged_structure), 188

test_can_get_repr() (in module test_can_get_value() (in module tests.blackfennec.type_system.test_map_type.MapTypeTestSuite method), 207), tests.blackfennec.structure.test_list), 200

test_can_get_repr() (in module test_can_get_value() (in module tests.blackfennec.type_system.test_number_type.NumberTypeTestSuite method), 208), tests.blackfennec.structure.test_map), 200

test_can_get_repr() (in module test_can_get_value() (in module tests.blackfennec.type_system.test_string_type.StringTypeTestSuite method), 209), tests.blackfennec.structure.test_boolean.BooleanTestSuite method), 199

test_can_get_resource_type_registry() (in module tests.blackfennec.extension_system.test_extensions_top), 178), tests.blackfennec.structure.test_structure.StructureTestMixin method), 202

test_can_get_root() (in module tests.blackfennec.layers.merge.test_integration_merged), 185), test_can_get_value_empty() (in module tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base), 183

test_can_get_structure() (in module tests.blackfennec.action_system.test_context), 171), test_can_get_value_empty() (in module tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base), 184

test_can_get_structure() (in module tests.blackfennec.layers.merge.test_merged_structure), 187), test_can_get_view_factory() (in module tests.blackfennec.extension_system.test_extension_api), 179

test_can_get_subject() (in module tests.blackfennec.layers.merge.test_merged_null), 187), test_can_get_view_factory_registry() (in module tests.blackfennec.extension_system.test_extension_api), 179

test_can_get_text() (in module tests.blackfennec.presentation_system.ui_service.test_message), 194), test_can_handle_uri() (in module tests.blackfennec.presentation_system.main_window.test_black_fennec), 192

test_can_get_timeout() (in module tests.blackfennec.presentation_system.ui_service.test_message), 194), test_load_document() (in module tests.blackfennec.presentation_system.main_window.test_document), 193

test_can_get_type() (in module tests.blackfennec.action_system.test_action), 171), test_load_extensions() (in module tests.blackfennec.extension_system.test_extension_service), 179

test_can_get_type_loader() (in module tests.blackfennec.extension_system.test_extensions_top), 178), test_load_extensions_with_complex_dependencies() (in module tests.blackfennec.extension_system.test_extension_service), 171

test_can_load_extensions_with_dependencies() (in module tests.blackfennec.extension_system.test_extensions_top), 178)

179
test_can_load_extensions_with_failing_extensions() (in module tests.blackfennec.extension_system.test_extension_system), 179
test_can_load_extensions_with_missing_dependencies() (in module tests.blackfennec.extension_system.test_extension_system), 180
test_can_make_index_optional() (tests.blackfennec.type_system.test_list_type.ListTypeTestSuite method), 206
test_can_make_index_required() (tests.blackfennec.type_system.test_list_type.ListTypeTestSuite method), 206
test_can_merge_merged_structure() (in module tests.blackfennec.layers.merge.test_merged_layer), 186
test_can_open() (in module tests.blackfennec.presentation_system.main_window.test_black_fennec), 192
test_can_open_file() (in module tests.blackfennec.presentation_system.main_window.test_black_fennec), 192
test_can_recognize_self() (tests.blackfennec.type_system.test_map_type.MapTypeTestSuite method), 207
test_can_recover_from_null_merge() (in module tests.blackfennec.layers.merge.test_merged_layer), 186
test_can_recover_from_null_merge_2() (in module tests.blackfennec.layers.merge.test_merged_layer), 186
test_can_redo() (in module tests.blackfennec.presentation_system.main_window.test_black_fennec), 192
test_can_redo() (in module tests.blackfennec.presentation_system.test_history_test_view), 196
test_can_register_action() (in module tests.blackfennec.action_system.test_action_registry), 171
test_can_register_extension() (in module tests.blackfennec.extension_system.test_extension_system), 179
test_can_register_message_overlay() (in module tests.blackfennec.presentation_system.ui_service.test_ui_service), 195
test_can_register_two_actions_with_same_type() (in module tests.blackfennec.action_system.test_action_registry), 171
test_can_register_view_factory() (in module tests.blackfennec.extension_system.test_view_factory), 181
test_can_remember_structure() (in module tests.blackfennec.layers.observable.test_observable), 179
test_can_remove_structure() (in module tests.blackfennec.layers.overlay.test_overlay), 189
test_can_remove_equal_numbers_from_map() (tests.blackfennec.structure.test_number.NumberTestSuite method), 201
test_can_remove_item() (in module tests.blackfennec.structure.test_map), 200
test_can_remove_item_item() (in module tests.blackfennec.structure.test_list), 200
test_can_request_preview() (tests.blackfennec.type_system.interpretation.test_specification.SpecificationTestSuite method), 205
test_can_reset_limits() (tests.blackfennec.type_system.test_number_type.TestNumberTypeTestSuite method), 208
test_can_resolve() (in module tests.blackfennec.layers.encapsulation_base.test_reference_encapsulation_base), 185
test_can_resolve_double_reference() (in module tests.blackfennec.layers.overlay.test_integration_overlay), 190
test_can_save() (in module tests.blackfennec.presentation_system.main_window.test_document), 193
test_can_save_as() (in module tests.blackfennec.presentation_system.main_window.test_document), 193
test_can_save_as_file() (in module tests.blackfennec.presentation_system.main_window.test_black_fennec), 192
test_can_save_and_load() (in module tests.blackfennec.presentation_system.main_window.test_black_fennec), 192
test_can_set_child_optional() (tests.blackfennec.type_system.test_list_type.ListTypeTestSuite method), 206
test_can_set_child_optional() (tests.blackfennec.type_system.test_map_type.MapTypeTestSuite method), 207
test_can_set_directory() (in module tests.blackfennec.presentation_system.main_window.test_black_fennec), 192
test_can_set_parent() (tests.blackfennec.structure.test_structure.StructureTestMixin method), 202
test_can_set_required_to_false() (tests.blackfennec.type_system.test_map_type.MapTypeTestSuite method), 207
test_can_set_required_to_true() (tests.blackfennec.type_system.test_map_type.MapTypeTestSuite method), 207
test_can_set_value() (in module tests.blackfennec.layers.observable.test_observable), 179

`tests.blackfennec.structure.test_list`), 200
`test_can_set_value()` (in module `tests.blackfennec.layers.observable.test_integration_observable_list`), 189
`test_can_set_value()` (in module `tests.blackfennec.structure.test_map`), 200
`test_can_set_value()` (in module `tests.blackfennec.structure.test_reference`), 202
`test_can_set_value()` (`tests.blackfennec.structure.test_boolean.BooleanTestSuite` method), 199
`test_can_set_value()` (`tests.blackfennec.structure.test_structure.StructureTestMixin` method), 202
`test_can_set_value_when_map_has_content()` (in module `tests.blackfennec.structure.test_map`), 201
`test_can_show_action_message()` (in module `tests.blackfennec.presentation_system.ui_service.test_ui_service`), 195
`test_can_show_simple_message()` (in module `tests.blackfennec.presentation_system.ui_service.test_ui_service`), 195
`test_can_tell_if_can_redo()` (in module `tests.blackfennec.presentation_system.test_history_service`), 196
`test_can_tell_that_child_is_optional()` (`tests.blackfennec.type_system.test_list_type.ListTypeTestSuite` method), 206
`test_can_tell_that_child_is_optional()` (`tests.blackfennec.type_system.test_map_type.MapTypeTestSuite` method), 207
`test_can_tell_that_child_is_required()` (`tests.blackfennec.type_system.test_list_type.ListTypeTestSuite` method), 206
`test_can_tell_that_child_is_required()` (`tests.blackfennec.type_system.test_map_type.MapTypeTestSuite` method), 207
`test_can_undo()` (in module `tests.blackfennec.presentation_system.main_window.test_black_fennec_view_controller`), 192
`test_can_undo()` (in module `tests.blackfennec.presentation_system.test_history_service`), 196
`test_can_undo_add_on_list()` (in module `tests.blackfennec.layers.observable.test_integration_observable_list`), 189
`test_can_undo_modification_on_number_in_list()` (in module `tests.blackfennec.layers.observable.test_integration_observable_list`), 189
`test_can_undo_modification_on_string()` (in module `tests.blackfennec.layers.observable.test_integration_observable_list`), 189
`test_can_undo_redo()` (in module `tests.blackfennec.presentation_system.test_history_service`), 196
`test_can_undo_redo_on_list()` (in module `tests.blackfennec.layers.observable.test_integration_observable_list`), 189
`test_can_undo_redo_on_map()` (in module `tests.blackfennec.layers.observable.test_integration_observable_list`), 189
`test_can_undo_redo_on_top_level_map()` (in module `tests.blackfennec.layers.observable.test_integration_observable_list`), 189
`test_can_visit_boolean()` (`tests.blackfennec.layers.encapsulation_base.test_base_factory_visitor` method), 181
`test_can_visit_list()` (`tests.blackfennec.layers.encapsulation_base.test_base_factory_visitor` method), 181
`test_can_visit_map()` (`tests.blackfennec.layers.encapsulation_base.test_base_factory_visitor` method), 181
`test_can_visit_number()` (`tests.blackfennec.layers.encapsulation_base.test_base_factory_visitor` method), 181
`test_can_visit_reference()` (`tests.blackfennec.layers.encapsulation_base.test_base_factory_visitor` method), 181
`test_can_visit_string()` (`tests.blackfennec.layers.encapsulation_base.test_base_factory_visitor` method), 181
`test_can_visit_structure()` (`tests.blackfennec.layers.encapsulation_base.test_base_factory_visitor` method), 181
`test_can_visit_structure()` (`tests.blackfennec.layers.overlay.test_overlay_factory_visitor.OverlayFactoryVisitor` method), 191
`test_cannot_attach_tab_twice()` (in module `tests.blackfennec.presentation_system.main_window.test_black_fennec_view_controller`), 192
`test_cannot_check_if_number_in_list()` (`tests.blackfennec.presentation_system.test_history_service` method), 201
`test_cannot_compare_merged_objects_with_different_values()` (in module `tests.blackfennec.layers.merge.test_merged_layer`), 186
`test_cannot_compare_merged_objects_with_standard_objects()` (in module `tests.blackfennec.layers.merge.test_merged_layer`), 186
`test_cannot_construct_with_none_null()` (in module `tests.blackfennec.layers.merge.test_merged_null`), 187
`test_cannot_deregister_action_twice()` (in module `tests.blackfennec.action_system.test_action_registry`), 171
`test_cannot_deregister_view_factory_if_not_registered()` (in module `tests.blackfennec.extension_system.test_view_factory`), 181

test_cannot_detach_tab_which_is_not_attached() test_create_document_registry() (in module tests.blackfennec.presentation_system.main_window.test_blackfennec_document_system.test_document_registry), 192 177

test_cannot_find_number_in_list() test_create_document_without_mime_type() (in module tests.blackfennec.structure.test_number.NumberTestSuite tests.blackfennec.document_system.test_document_factory method), 201 176

test_cannot_get_factory_if_not_exists() (in test_create_interpretation() module tests.blackfennec.extension_system.test_view_factory.tests.blackfennec.type_system.interpretation.test_interpretation.In method), 203 181

test_cannot_handle_uri() (in module test_create_interpreter() (in module tests.blackfennec.presentation_system.main_window.test_blackfennec_type_system.interpretation.test_interpretation_s method), 204 192

test_cannot_redo() (in module test_create_mime_type_registry() tests.blackfennec.presentation_system.main_window.test_blackfennec_document_system.mime_type.test_mime_type_r method), 174 192

test_cannot_set_none_child_optionality() test_create_navigation_proxy() (tests.blackfennec.type_system.test_list_type.ListTypeTestSuite tests.blackfennec.presentation_system.navigation_service.test_na method), 206 194

test_cannot_set_none_child_optionality() test_create_navigation_service() (tests.blackfennec.type_system.test_map_type.MapTypeTestSuite tests.blackfennec.presentation_system.navigation_service.test_na method), 207 194

test_cannot_set_parent() (in module test_create_presenter_registry() tests.blackfennec.layers.merge.test_merged_structure), (tests.blackfennec.extension_system.test_presenter_registry.Prese method), 180 188

test_cannot_set_required_for_none_property() test_create_resource_type_registry() (tests.blackfennec.type_system.test_map_type.MapTypeTestSuite tests.blackfennec.document_system.resource_type.test_resource_ method), 207 175

test_cannot_set_value() (in module test_create_type() (in module tests.blackfennec.layers.merge.test_merged_list), tests.blackfennec.type_system.test_type_parser), 187 209

test_cannot_set_value() (in module test_create_type_registry() tests.blackfennec.layers.merge.test_merged_map), (tests.blackfennec.type_system.test_type_registry.TypeRegistryTes method), 209 187

test_cannot_set_value() (in module test_default() (in module tests.blackfennec.layers.merge.test_merged_null), tests.blackfennec.type_system.test_null_type), 207 187

test_cannot_set_value() (in module test_default() (tests.blackfennec.type_system.test_boolean_type.TestBoo tests.blackfennec.layers.merge.test_merged_structure), method), 205 188

test_cannot_set_value() (in module test_default() (tests.blackfennec.type_system.test_number_type.TestNum tests.blackfennec.layers.merge.test_merged_structure), method), 208 188

test_cannot_undo() (in module tests.blackfennec.presentation_system.main_window.test_blackfennec_document_system.mime_type.test_mime_type_r method), 208 192

test_coverage_getter_list_full_coverage() test_deregister_mime_type() (tests.blackfennec.document_system.mime_type.test_mime_type_r method), 174 206

test_coverage_getter_list_half_coverage() test_deregister_resource_type() (tests.blackfennec.document_system.resource_type.test_resource_ method), 175 206

test_coverage_getter_simple() (in module test_deregister_view() (tests.blackfennec.extension_system.test_presenter_registry.Prese method), 180 204

test_create_document() (in module tests.blackfennec.type_system.interpretation.test_off method), 209 176

test_create_document() (in module tests.blackfennec.document_system.test_document_factory method), 209 176

test_create_serialize() (in module tests.blackfennec.document_system.mime_type.json.test_structure method), 209 176

173
test_determine_mimetype_of_file() (tests.blackfennec.document_system.mime_type.test_mime_type_blackfennec.type_system.interpretation.test_coverage.Coverage method), 174
test_determine_mimetype_of_unknown() (tests.blackfennec.document_system.mime_type.test_mime_type_blackfennec.type_system.interpretation.test_coverage.Coverage method), 174
test_determine_mimetype_online() (tests.blackfennec.document_system.resource_type.protocols.test_blackfennec.layers.resource_type_https_resources_type_blackfennec.type_system.interpretation.test_coverage.Coverage method), 175
test_determine_resource_type_of_file() (tests.blackfennec.document_system.resource_type.test_resource_blackfennec.type_system.interpretation.test_coverage.Coverage method), 175
test_determine_resource_type_of_http() (tests.blackfennec.document_system.resource_type.test_resource_blackfennec.type_system.interpretation.test_coverage.Coverage method), 175
test_determine_resource_type_of_https() (tests.blackfennec.document_system.resource_type.test_resource_blackfennec.type_system.interpretation.test_coverage.Coverage method), 175
test_dispatch_change_notification() (in module tests.blackfennec.layers.encapsulation_base.test_list_encapsulated_blackfennec.type_system.interpretation.test_coverage.Coverage 183
test_dispatch_change_notification() (in module tests.blackfennec.layers.encapsulation_base.test_map_encapsulated_blackfennec.type_system.interpretation.test_offer), 184
test_dispatch_change_notification() (in module tests.blackfennec.util.test_change_notification_dispatch_mixed_blackfennec.document_system.mime_type.json.test_json_mimetype_blackfennec.type_system.interpretation.test_offer), 210
test_does_append_to_history() (in module tests.blackfennec.layers.observable.test_observable_base), 189
test_does_not_remember_reference() (in module tests.blackfennec.layers.overlay.test_overlay), 190
test_does_register_document() (tests.blackfennec.document_system.test_document.TestDocument_blackfennec.type_system.test_integration_type_loader), method), 176
test_double_encapsulation_1() (tests.blackfennec.layers.encapsulation_base.test_base_factory_blackfennec.type_system.test_integration_type_loader), method), 181
test_double_encapsulation_2() (tests.blackfennec.layers.encapsulation_base.test_base_factory_blackfennec.type_system.test_integration_type_loader), method), 181
test_double_encapsulation_3() (tests.blackfennec.layers.encapsulation_base.test_base_factory_blackfennec.type_system.test_integration_type_loader), method), 181
test_double_encapsulation_4() (tests.blackfennec.layers.encapsulation_base.test_base_factory_blackfennec.type_system.test_integration_type_loader), method), 182
test_double_encapsulation_5() (tests.blackfennec.layers.encapsulation_base.test_base_factory_blackfennec.type_system.test_integration_type_loader), method), 182
test_double_encapsulation_6() (tests.blackfennec.layers.encapsulation_base.test_base_factory_blackfennec.type_system.test_integration_type_loader), method), 182
method), 182
test_encapsulated_can_visit_1() (tests.blackfennec.layers.encapsulation_base.test_base_factory_blackfennec.type_system.test_integration_type_loader), method), 182
test_encapsulated_can_visit_2() (tests.blackfennec.layers.encapsulation_base.test_base_factory_blackfennec.type_system.test_integration_type_loader), method), 182
test_encapsulated_can_visit_3() (tests.blackfennec.layers.encapsulation_base.test_base_factory_blackfennec.type_system.test_integration_type_loader), method), 182
test_encapsulated_can_visit_4() (tests.blackfennec.layers.encapsulation_base.test_base_factory_blackfennec.type_system.test_integration_type_loader), method), 182
test_encapsulated_can_visit_5() (tests.blackfennec.layers.encapsulation_base.test_base_factory_blackfennec.type_system.test_integration_type_loader), method), 182
test_encapsulated_can_visit_6() (tests.blackfennec.layers.encapsulation_base.test_base_factory_blackfennec.type_system.test_integration_type_loader), method), 182
test_equal_coverages_equality() (tests.blackfennec.type_system.interpretation.test_coverage.Coverage method), 203
test_equal_offers_equality() (in module tests.blackfennec.type_system.interpretation.test_offer), 204
test_export_structure() (in module tests.blackfennec.document_system.mime_type.json.test_json_mimetype_blackfennec.type_system.interpretation.test_offer), 172
test_failing_navigate_in_map() (in module tests.blackfennec.structure.reference_navigation.test_sibling_offspring_blackfennec.type_system.test_integration_type_loader), 198
test_for_subclass_property() (in module tests.blackfennec.type_system.test_integration_type_loader), 205
test_for_superclass_property() (in module tests.blackfennec.type_system.test_integration_type_loader), 205
test_generic_overlay_subject() (tests.blackfennec.layers.overlay_factory_visitor_blackfennec.type_system.test_integration_type_loader), method), 191
test_get_app_id() (in module tests.blackfennec.util.test_meta_info), 211
test_get_authors() (in module tests.blackfennec.util.test_meta_info), 211
test_get_content_cached() (tests.blackfennec.layers.encapsulation_base.test_base_factory_blackfennec.type_system.test_integration_type_loader), method), 176
test_get_copy_right() (in module tests.blackfennec.util.test_meta_info), 211
test_get_description() (in module tests.blackfennec.util.test_meta_info), 211
test_get_document() (tests.blackfennec.type_system.test_integration_type_loader), method), 176

<code>tests.blackfennec.document_system.test_document_system</code>	<code>test_greater_equal_equal_elements()</code>	(<code>tests.blackfennec.util.test_comparable.ComparableTestSuite</code>)
177		
<code>test_get_document_for_created_document()</code> (in module <code>tests.blackfennec.document_system.test_document_system</code>)	<code>test_greater_equal_greater_element()</code>	(<code>tests.blackfennec.util.test_comparable.ComparableTestSuite</code>)
176		
<code>test_get_home_page()</code> (in module <code>tests.blackfennec.util.test_meta_info</code>)	<code>test_greater_equal_lower_element()</code>	(<code>tests.blackfennec.util.test_comparable.ComparableTestSuite</code>)
211		
<code>test_get_icon_path()</code> (in module <code>tests.blackfennec.util.test_meta_info</code>)	<code>test_greater_than_equal_elements()</code>	(<code>tests.blackfennec.util.test_comparable.ComparableTestSuite</code>)
211		
<code>test_get_issue_page()</code> (in module <code>tests.blackfennec.util.test_meta_info</code>)	<code>test_greater_than_greater_element()</code>	(<code>tests.blackfennec.util.test_comparable.ComparableTestSuite</code>)
211		
<code>test_get_license()</code> (in module <code>tests.blackfennec.util.test_meta_info</code>)	<code>test_greater_than_lower_element()</code>	(<code>tests.blackfennec.util.test_comparable.ComparableTestSuite</code>)
211		
<code>test_get_name()</code> (in module <code>tests.blackfennec.util.test_meta_info</code>)	<code>test_has_create_instance_interface()</code> (in module <code>tests.blackfennec.type_system.test_type_base</code>)	209
211		
<code>test_get_new_value()</code> (in module <code>tests.blackfennec.util.test_change_notification</code>)	<code>test_hash_equal_tokens()</code> (in module <code>tests.blackfennec.structure.reference_navigation.test_child_navigation</code>)	196
210		
<code>test_get_old_value()</code> (in module <code>tests.blackfennec.util.test_change_notification</code>)	<code>test_hash_not_equal_tokens()</code> (in module <code>tests.blackfennec.structure.reference_navigation.test_child_navigation</code>)	196
210		
<code>test_get_release_notes()</code> (in module <code>tests.blackfennec.util.test_meta_info</code>)	<code>test_hash_of_navigators_with_different_offset_are_not_equal()</code> (in module <code>tests.blackfennec.structure.reference_navigation.test_child_navigation</code>)	197
211		
<code>test_get_representation()</code> (in module <code>tests.blackfennec.structure.reference_navigation.test_child_navigation</code>)	<code>test_hash_of_navigators_with_different_offset_are_not_equal()</code> (in module <code>tests.blackfennec.structure.reference_navigation.test_child_navigation</code>)	198
196		
<code>test_get_representation()</code> (in module <code>tests.blackfennec.structure.reference_navigation.test_child_navigation</code>)	<code>test_hash_of_navigators_with_same_offset_are_equal()</code> (in module <code>tests.blackfennec.structure.reference_navigation.test_child_navigation</code>)	198
197		
<code>test_get_representation()</code> (in module <code>tests.blackfennec.structure.reference_navigation.test_child_navigation</code>)	<code>test_hash_of_navigators_with_same_offset_are_equal()</code> (in module <code>tests.blackfennec.structure.reference_navigation.test_child_navigation</code>)	199
198		
<code>test_get_representation()</code> (in module <code>tests.blackfennec.structure.reference_navigation.test_child_navigation</code>)	<code>test_import_structure()</code> (in module <code>tests.blackfennec.document_system.mime_type.json.test_json_mime_type</code>)	172
198		
<code>test_get_representation()</code> (in module <code>tests.blackfennec.structure.reference_navigation.test_child_navigation</code>)	<code>test_instance_in_inherited()</code> (in module <code>tests.blackfennec.type_system.test_integration_type_loader</code>)	205
198		
<code>test_get_representation()</code> (in module <code>tests.blackfennec.structure.reference_navigation.test_child_navigation</code>)	<code>test_inherits_maximum()</code> (in module <code>tests.blackfennec.type_system.test_integration_type_loader</code>)	206
199		
<code>test_get_root_returns_merged_structure()</code> (in module <code>tests.blackfennec.layers.merge.test_integration_type_loader</code>)	<code>test_is_absolute_json_pointer()</code> (in module <code>tests.blackfennec.document_system.mime_type.json.test_json_mime_type</code>)	183
185		
<code>test_get_summary()</code> (in module <code>tests.blackfennec.util.test_meta_info</code>)	<code>test_is_covered_false()</code>	(<code>tests.blackfennec.type_system.interpretation.test_coverage.Coverage</code>)
211		
<code>test_get_value()</code> (in module <code>tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base</code>)		
183		
<code>test_get_value()</code> (in module <code>tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base</code>)		
184		
<code>test_get_version()</code> (in module <code>tests.blackfennec.util.test_meta_info</code>)		
211		

test_is_covered_true() (tests.blackfennec.type_system.interpretation.test_coverage.CoverageTestSuite method), 203

test_is_json_reference() (in module tests.blackfennec.document_system.mime_type.json.test_json_serialization.merge.test_merged_layer), 173

test_is_json_reference_with_no_json_reference() (in module tests.blackfennec.document_system.mime_type.json.test_json_serialization.merge.test_merged_layer), 173

test_is_merge_two_merged_maps_with_null_properties() (in module tests.blackfennec.document_system.mime_type.json.test_json_serialization.merge.test_merged_layer), 185

test_is_relative_json_pointer() (in module tests.blackfennec.document_system.mime_type.json.test_json_serialization.merge.test_merged_layer), 172

test_load_resource() (tests.blackfennec.document_system.resource_type.protocol.test_blackfennec_layers_resource_type_protocol.ResourceTypeTestSuite method), 175

test_load_resource_absolute_path() (tests.blackfennec.document_system.resource_type.protocol.test_blackfennec_layers_resource_type_protocol.ResourceTypeTestSuite method), 175

test_load_resource_relative_path() (tests.blackfennec.document_system.resource_type.protocol.test_blackfennec_layers_resource_type_protocol.ResourceTypeTestSuite method), 175

test_logs_on_remove_item_not_existing() (in module tests.blackfennec.structure.test_map), 201

test_lower_equal_equal_elements() (tests.blackfennec.util.test_comparable.ComparableTestSuite method), 210

test_lower_equal_greater_element() (tests.blackfennec.util.test_comparable.ComparableTestSuite method), 210

test_lower_equal_lower_element() (tests.blackfennec.util.test_comparable.ComparableTestSuite method), 210

test_lower_than_equal() (in module tests.blackfennec.type_system.interpretation.test_offer), 204

test_lower_than_equal() (tests.blackfennec.type_system.interpretation.test_coverage.CoverageTestSuite method), 203

test_lower_than_lower_and_greater() (tests.blackfennec.type_system.interpretation.test_coverage.CoverageTestSuite method), 203

test_lower_than_with_different_subject() (in module tests.blackfennec.type_system.interpretation.test_offer), 204

test_maximum_default() (tests.blackfennec.type_system.test_number_type.TestNumberType method), 208

test_merge_merged_overlay() (in module tests.blackfennec.layers.merge.test_merged_layer), 186

test_merge_null() (in module tests.blackfennec.type_system.test_integration_type_loader), 206

test_merge_super_first_level() (in module tests.blackfennec.layers.merge.test_merged_layer), 186

test_merge_super_second_level() (in module tests.blackfennec.layers.merge.test_merged_layer), 186

test_merged_map_null_property() (in module tests.blackfennec.layers.merge.test_merged_layer), 185

test_merged_parent_is_self() (in module tests.blackfennec.layers.merge.test_merged_layer), 185

test_merges_properties() (in module tests.blackfennec.layers.merge.test_merged_layer), 186

test_merges_recursively() (in module tests.blackfennec.type_system.test_integration_type_loader), 206

test_minimum_default() (tests.blackfennec.type_system.test_boolean_type.TestBooleanType method), 205

test_minimum_default() (tests.blackfennec.type_system.test_number_type.TestNumberType method), 208

test_must_override_execute() (in module tests.blackfennec.action_system.test_action), 171

test_navigate() (in module tests.blackfennec.structure.reference_navigation.test_parent_navigation), 198

test_navigate() (in module tests.blackfennec.structure.reference_navigation.test_root_navigation), 198

test_navigate() (in module tests.blackfennec.structure.reference_navigation.test_sibling_offspring_navigation), 198

test_navigate() (tests.blackfennec.presentation_system.navigation_service.NavigationService method), 194

test_navigate() (tests.blackfennec.presentation_system.navigation_service.NavigationService method), 194

test_navigate_from_child_without_parent() (in module tests.blackfennec.structure.reference_navigation.test_index_navigation), 197

test_navigate_invalid_type() (in module tests.blackfennec.structure.reference_navigation.test_child_navigation), 197

196
test_navigate_list() (in module test_navigators_with_different_offset_are_not_equal()
tests.blackfennec.structure.reference_navigation.test_child_navigation), 196

196
test_navigate_list() (in module test_navigators_with_different_offset_are_not_equal()
tests.blackfennec.structure.reference_navigation.test_index_navigation), 196

197
test_navigate_list_invalid_index() (in module test_navigators_with_same_offset_are_equal()
tests.blackfennec.structure.reference_navigation.test_child_navigation), 197

197
test_navigate_map() (in module test_navigators_with_same_offset_are_equal()
tests.blackfennec.structure.reference_navigation.test_child_navigation), 197

197
test_navigate_map() (in module test_not_equal_coverages_equality()
tests.blackfennec.structure.reference_navigation.test_index_navigation), 197

197
test_navigate_map_invalid_key() (in module test_not_equal_equal_elements()
tests.blackfennec.structure.reference_navigation.test_child_navigation), 197

test_navigate_without_presenter() test_not_equal_unequal_elements()
(tests.blackfennec.presentation_system.navigation_service.test_blackfennec_util_navigation_service_test_suite), 194

test_navigation_destination_is_argument() test_notifies_on_item_add() (in module
tests.blackfennec.type_system.interpretation.test_interpretation_blackfennec_test_suite), 203

test_navigation_in_doubly_merged_object() (in module test_notifies_on_item_add() (in module
tests.blackfennec.structure.test_list), 200

186 test_notifies_on_item_add() (in module
tests.blackfennec.structure.test_map), 201

test_navigation_in_merged_object() (in module test_notifies_on_item_add() (in module
tests.blackfennec.layers.merge.test_merged_layer), 186

186 test_notifies_on_item_add() (in module
tests.blackfennec.structure.test_list), 200

test_navigation_in_phantom_parents() (in module test_notifies_on_item_add() (in module
tests.blackfennec.layers.merge.test_merged_layer), 186

186 test_notifies_on_item_add() (in module
tests.blackfennec.structure.test_map), 201

test_navigation_request_is_dispatched() test_notifies_on_value_change() (in module
tests.blackfennec.type_system.interpretation.test_interpretation_blackfennec_test_suite), 203

203 test_notifies_on_value_change() (in module
tests.blackfennec.structure.test_reference), 202

test_navigation_sender_is_interpretation() test_notifies_on_value_change() (in module
tests.blackfennec.type_system.interpretation.test_interpretation_blackfennec_test_suite), 203

203 test_notifies_on_value_change() (in module
tests.blackfennec.type_system.test_boolean_type.TestBooleanType), 205

test_navigator_equality() (in module test_omitted_default() (in module
tests.blackfennec.type_system.test_number_type.TestNumberType), 197

197 test_omitted_default() (in module
tests.blackfennec.type_system.test_number_type.TestNumberType), 208

test_navigator_hash() (in module test_omitted_maximum() (in module
tests.blackfennec.type_system.test_number_type.TestNumberType), 197

197 test_omitted_maximum() (in module
tests.blackfennec.type_system.test_number_type.TestNumberType), 208

test_navigator_inequality() (in module test_omitted_minimum() (in module
tests.blackfennec.type_system.test_number_type.TestNumberType), 197

197 test_omitted_minimum() (in module
tests.blackfennec.type_system.test_number_type.TestNumberType), 208

test_navigators_are_equal() (in module test_only_deregisters_correct_action() (in
tests.blackfennec.structure.reference_navigation.test_child_navigation), 197

197 test_only_deregisters_correct_action() (in
tests.blackfennec.action_system.test_action_registry), 171

test_navigators_are_not_equal() (in module test_only_merge_with_same_type() (in module
tests.blackfennec.structure.reference_navigation.test_child_navigation), 197

197 test_only_merge_with_same_type() (in module
tests.blackfennec.layers.merge.test_integration_merged), 171

test_replace_encapsulated_item() (in module tests.blackfennec.layers.encapsulation_base.test_encapsulation_base), 205
 test_replace_item() (in module tests.blackfennec.layers.encapsulation_base.test_encapsulation_base), 183
 test_replace_item() (in module tests.blackfennec.layers.encapsulation_base.test_encapsulation_base), 183
 test_replace_item() (in module tests.blackfennec.layers.encapsulation_base.test_encapsulation_base), 184
 test_replace_item() (in module tests.blackfennec.structure.test_map), 201
 test_replace_item_notifies_only_once() (in module tests.blackfennec.structure.test_map), 201
 test_repr() (in module tests.blackfennec.layers.merge.test_merged_list), 187
 test_repr() (in module tests.blackfennec.layers.merge.test_merged_map), 187
 test_repr() (in module tests.blackfennec.layers.merge.test_merged_null), 187
 test_repr() (in module tests.blackfennec.type_system.test_null_type), 207
 test_repr() (tests.blackfennec.type_system.test_boolean_type.TestBooleanType method), 205
 test_representation() (in module tests.blackfennec.type_system.interpretation.test_offer), 204
 test_representation() (tests.blackfennec.structure.test_boolean.BooleanTestSuite method), 199
 test_representation() (tests.blackfennec.type_system.interpretation.test_specification.SpecificationTestSuite method), 205
 test_required_properties() (tests.blackfennec.type_system.test_map_type.MapTypeTestSuite method), 207
 test_resolve() (in module tests.blackfennec.structure.test_reference), 202
 test_respects_inheritance_hierarchy() (in module tests.blackfennec.type_system.interpretation.test_offer), 204
 test_returns_copy_of_internal_list() (tests.blackfennec.type_system.test_type_registry.TypeRegistryTestSuite method), 209
 test_returns_empty_list_by_default() (in module tests.blackfennec.action_system.test_action_registry), 171
 test_reuses_structure_for_expected() (tests.blackfennec.type_system.test_boolean_type.TestBooleanType method), 205
 test_root_getter_1() (tests.blackfennec.layers.encapsulation_base.test_encapsulation_base), 182
 test_root_getter_2() (tests.blackfennec.layers.encapsulation_base.test_encapsulation_base), 182
 test_root_getter_3() (tests.blackfennec.layers.encapsulation_base.test_encapsulation_base), 182
 test_root_getter_4() (tests.blackfennec.layers.encapsulation_base.test_encapsulation_base), 182
 test_root_getter_5() (tests.blackfennec.layers.encapsulation_base.test_encapsulation_base), 182
 test_root_getter_6() (tests.blackfennec.layers.encapsulation_base.test_encapsulation_base), 183
 test_save() (tests.blackfennec.document_system.test_document.TestDocument method), 176
 test_save_all() (in module tests.blackfennec.presentation_system.main_window.test_black_fennec), 192
 test_serialize() (in module tests.blackfennec.document_system.mime_type.json.test_structure), 173
 test_serialize_json_reference() (in module tests.blackfennec.document_system.mime_type.json.test_json_reference), 173
 test_serialize_relative_pointer() (in module tests.blackfennec.document_system.mime_type.json.test_json_pointer), 172
 test_set_expected_to_false() (tests.blackfennec.type_system.test_boolean_type.TestBooleanType method), 205
 test_set_expected_to_true() (tests.blackfennec.type_system.test_boolean_type.TestBooleanType method), 205
 test_set_maximum() (tests.blackfennec.type_system.test_number_type.TestNumberType method), 208
 test_set_minimum() (tests.blackfennec.type_system.test_number_type.TestNumberType method), 208
 test_set_navigation_service() (tests.blackfennec.type_system.interpretation.test_interpretation.InterpretationTestSuite method), 204
 test_set_value() (in module tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base), 184
 test_set_value() (in module tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base), 184
 test_skip_unnecessary_dispatch_of_change_notification() (in module tests.blackfennec.util.test_change_notification_dispatch), 171

210 method), 210

test_specificity_getter() (in module test_visit_caches_class() tests.blackfennec.type_system.interpretation.test_offer), (tests.blackfennec.layers.encapsulation_base.test_base_factory_v method), 182

test_structure_getter() test_visit_caches_class() (tests.blackfennec.type_system.interpretation.test_interpretation.test_blackfennec.TestSuite.overlay.test_overlay_factory_visitor.Over method), 204 method), 191

test_subject_getter() (in module test_visit_list() (tests.blackfennec.util.test_deep_copy.TestDeepCopy tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base), 184 test_visit_map() (tests.blackfennec.util.test_deep_copy.TestDeepCopyV method), 211

test_subject_getter() (in module tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base), 184 (tests.blackfennec.util.test_deep_copy.TestDeepCopyVisitorTestSu method), 211

test_subject_getter() (in module tests.blackfennec.type_system.interpretation.test_offer), 204 test_visit_reference() (tests.blackfennec.util.test_deep_copy.TestDeepCopyVisitorTestSu method), 211

test_subject_getter_1() (tests.blackfennec.layers.encapsulation_base.test_test_visitor_sharing() (tests.blackfennec.util.test_deep_copy.TestDeepCopyVisitorTestSu method), 183 method), 211

test_subject_getter_2() (tests.blackfennec.layers.encapsulation_base.test_test_visitor_sharing() (tests.blackfennec.util.test_deep_copy.TestDeepCopyVisitorTestSu method), 183 method), 211

test_subject_getter_3() (tests.blackfennec.layers.encapsulation_base.test_test_visitor_sharing() (tests.blackfennec.util.test_deep_copy.TestDeepCopyVisitorTestSu method), 183 method), 211

test_subject_getter_4() (tests.blackfennec.layers.encapsulation_base.test_test_visitor_sharing() (tests.blackfennec.util.test_deep_copy.TestDeepCopyVisitorTestSu method), 183 tests.blackfennec.type_system.test_null_type), 207

test_subject_getter_5() (tests.blackfennec.layers.encapsulation_base.test_test_visitor_sharing() (tests.blackfennec.util.test_deep_copy.TestDeepCopyVisitorTestSu method), 183 test_visitor() (tests.blackfennec.type_system.test_number_type.TestNum method), 205

test_subject_getter_6() (tests.blackfennec.layers.encapsulation_base.test_test_visitor_sharing() (tests.blackfennec.util.test_deep_copy.TestDeepCopyVisitorTestSu method), 183 test_visitor_allows_exactly() (tests.blackfennec.type_system.test_number_type.TestNumberType method), 205

test_throws_error_on_deserialization_of_unknown_type() (tests.blackfennec.type_system.test_number_type.TestNumberType (in module tests.blackfennec.document_system.mime_type.json_mime_type.structure_serializer), 173 test_visitor_checks_maximum() (tests.blackfennec.type_system.test_number_type.TestNumberType method), 205

test_throws_error_on_serialization_of_unknown_type() (tests.blackfennec.type_system.test_number_type.TestNumberType (in module tests.blackfennec.document_system.mime_type.json_mime_type.structure_serializer), 174 test_visitor_checks_minimum() (tests.blackfennec.type_system.test_number_type.TestNumberType method), 205

test_throws_on_remove_item_not_existing() (in module tests.blackfennec.structure.test_map), 201 (tests.blackfennec.type_system.test_boolean_type.TestBooleanType method), 205

test_type_covers_good_instance() (in module tests.blackfennec.type_system.test_integration_type_loader), 206 test_visitor_expected_false() (tests.blackfennec.type_system.test_boolean_type.TestBooleanType method), 205

test_type_from_json() (in module tests.blackfennec.type_system.test_type_parser), 209 TestBooleanType (class in tests.blackfennec.type_system.test_boolean_type), 205

test_type_getter() (in module tests.blackfennec.type_system.interpretation.test_offer), 204 TestDeepCopyVisitorTestSuite (class in tests.blackfennec.util.test_deep_copy), 210

test_visit_boolean() (tests.blackfennec.util.test_deep_copy.TestDeepCopyVisitorTestSuite (class in tests.blackfennec.document_system.test_document), 176 TestDocument (class in tests.blackfennec.document_system.test_document), 176

TestNumberType (class in tests.blackfennec.util.test_deep_copy), 210

<i>tests.blackfennec.type_system.test_number_type</i>),	<i>tests.blackfennec.extension_system.test_extension_api</i>
208	module, 177
tests	<i>tests.blackfennec.extension_system.test_extension_registry</i>
module, 214	module, 179
tests.blackfennec	<i>tests.blackfennec.extension_system.test_extension_service</i>
module, 212	module, 179
tests.blackfennec.action_system	<i>tests.blackfennec.extension_system.test_presenter_registry</i>
module, 171	module, 180
tests.blackfennec.action_system.test_action	<i>tests.blackfennec.extension_system.test_view_factory</i>
module, 170	module, 180
tests.blackfennec.action_system.test_action_registry	<i>tests.blackfennec.extension_system.test_view_factory_registry</i>
module, 171	module, 180
tests.blackfennec.action_system.test_context	<i>tests.blackfennec.layers</i>
module, 171	module, 191
tests.blackfennec.document_system	<i>tests.blackfennec.layers.encapsulation_base</i>
module, 177	module, 185
tests.blackfennec.document_system.mime_type	<i>tests.blackfennec.layers.encapsulation_base.test_base_factory</i>
module, 174	module, 181
tests.blackfennec.document_system.mime_type.jsontests	<i>tests.blackfennec.layers.encapsulation_base.test_encapsulation_base</i>
module, 174	module, 182
tests.blackfennec.document_system.mime_type.jsontests	<i>tests.blackfennec.layers.encapsulation_base.test_list_encapsulation_base</i>
module, 172	module, 183
tests.blackfennec.document_system.mime_type.jsontests	<i>tests.blackfennec.layers.encapsulation_base.test_map_encapsulation_base</i>
module, 172	module, 184
tests.blackfennec.document_system.mime_type.jsontests	<i>tests.blackfennec.layers.encapsulation_base.test_reference_encapsulation_base</i>
module, 173	module, 185
tests.blackfennec.document_system.mime_type.jsontests	<i>tests.blackfennec.layers.merge</i>
module, 173	module, 188
tests.blackfennec.document_system.mime_type.jsontests	<i>tests.blackfennec.layers.merge.test_integration_merged</i>
module, 174	module, 185
tests.blackfennec.document_system.mime_type.jsontests	<i>tests.blackfennec.layers.merge.test_merged_layer</i>
module, 174	module, 186
tests.blackfennec.document_system.resource_type	<i>tests.blackfennec.layers.merge.test_merged_list</i>
module, 176	module, 187
tests.blackfennec.document_system.resource_type	<i>tests.blackfennec.layers.merge.test_merged_map</i>
module, 175	module, 187
tests.blackfennec.document_system.resource_type	<i>tests.blackfennec.layers.merge.test_merged_null</i>
module, 174	module, 187
tests.blackfennec.document_system.resource_type	<i>tests.blackfennec.layers.merge.test_merged_optional</i>
module, 175	module, 187
tests.blackfennec.document_system.resource_type	<i>tests.blackfennec.layers.merge.test_merged_structure</i>
module, 175	module, 187
tests.blackfennec.document_system.resource_type	<i>tests.blackfennec.layers.observable</i>
module, 175	module, 190
tests.blackfennec.document_system.test_document	<i>tests.blackfennec.layers.observable.test_integration_observable</i>
module, 176	module, 188
tests.blackfennec.document_system.test_document	<i>tests.blackfennec.layers.observable.test_observable</i>
module, 176	module, 189
tests.blackfennec.document_system.test_document	<i>tests.blackfennec.layers.observable.test_observable_base</i>
module, 177	module, 189
tests.blackfennec.extension_system	<i>tests.blackfennec.layers.observable.test_observable_entry</i>
module, 181	module, 190
tests.blackfennec.extension_system.test_extensions	<i>tests.blackfennec.layers.observable.test_observable_factor</i>
module, 177	module, 190

tests.blackfennec.layers.overlay module, 191	tests.blackfennec.structure.test_map module, 200
tests.blackfennec.layers.overlay.test_integration_overlay module, 190	tests.blackfennec.structure.test_null module, 201
tests.blackfennec.layers.overlay.test_overlay module, 190	tests.blackfennec.structure.test_number module, 201
tests.blackfennec.layers.overlay.test_overlay_base module, 190	tests.blackfennec.structure.test_reference module, 202
tests.blackfennec.layers.overlay.test_overlay_base_overlay module, 191	tests.blackfennec.structure.test_string module, 202
tests.blackfennec.presentation_system module, 196	tests.blackfennec.structure.test_structure module, 202
tests.blackfennec.presentation_system.main_window_test module, 194	tests.blackfennec.type_system module, 209
tests.blackfennec.presentation_system.main_window_test_blackfennec_type_system_dialog module, 191	tests.blackfennec.type_system.dialog.interpretation module, 205
tests.blackfennec.presentation_system.main_window_test_blackfennec_type_system_dialog_interpretation.test_coverage module, 193	tests.blackfennec.type_system.dialog.interpretation.test_coverage module, 203
tests.blackfennec.presentation_system.navigation_system_proxy module, 194	tests.blackfennec.type_system.dialog.interpretation.test_integration module, 203
tests.blackfennec.presentation_system.navigation_system_proxy.interpretation.test_interpretation module, 194	tests.blackfennec.type_system.dialog.interpretation.test_interpretation module, 203
tests.blackfennec.presentation_system.navigation_system_proxy.interpretation.test_interpretation.test_interpretation module, 194	tests.blackfennec.type_system.dialog.interpretation.test_interpretation.test_interpretation module, 204
tests.blackfennec.presentation_system.test_history module, 195	tests.blackfennec.type_system.dialog.interpretation.test_offer module, 204
tests.blackfennec.presentation_system.ui_services module, 195	tests.blackfennec.type_system.dialog.interpretation.test_specific module, 205
tests.blackfennec.presentation_system.ui_services_blackfennec_type_system.test_boolean_type module, 194	tests.blackfennec.type_system.dialog.interpretation.test_boolean_type module, 205
tests.blackfennec.presentation_system.ui_services_blackfennec_type_system.test_integration_type_loader module, 195	tests.blackfennec.type_system.dialog.interpretation.test_integration_type_loader module, 205
tests.blackfennec.structure module, 203	tests.blackfennec.type_system.test_list_type module, 206
tests.blackfennec.structure.reference_navigation_base module, 199	tests.blackfennec.type_system.test_map_type module, 207
tests.blackfennec.structure.reference_navigation_base_blackfennec_type_system.test_null_type module, 196	tests.blackfennec.type_system.test_map_type_blackfennec_type_system.test_null_type module, 207
tests.blackfennec.structure.reference_navigation_base_blackfennec_type_system.test_number_type module, 197	tests.blackfennec.type_system.test_map_type_blackfennec_type_system.test_number_type module, 208
tests.blackfennec.structure.reference_navigation_base_blackfennec_type_system.test_string_type module, 197	tests.blackfennec.type_system.test_map_type_blackfennec_type_system.test_string_type module, 208
tests.blackfennec.structure.reference_navigation_base_blackfennec_type_system.test_type_base module, 198	tests.blackfennec.type_system.test_map_type_blackfennec_type_system.test_type_base module, 209
tests.blackfennec.structure.reference_navigation_base_blackfennec_type_system.test_type_parser module, 198	tests.blackfennec.type_system.test_map_type_blackfennec_type_system.test_type_parser module, 209
tests.blackfennec.structure.reference_navigation_base_blackfennec_type_system.test_type_registry module, 198	tests.blackfennec.type_system.test_map_type_blackfennec_type_system.test_type_registry module, 209
tests.blackfennec.structure.reference_navigation_base_blackfennec_type_system.test_type_registry.test_type_registry module, 199	tests.blackfennec.type_system.test_map_type_blackfennec_type_system.test_type_registry.test_type_registry module, 212
tests.blackfennec.structure.test_boolean module, 199	tests.blackfennec.util.test_change_notification module, 209
tests.blackfennec.structure.test_list module, 200	tests.blackfennec.util.test_change_notification_dispatch module, 210

tests.blackfennec.util.test_comparable module, 210

tests.blackfennec.util.test_deep_copy module, 210

tests.blackfennec.util.test_integration_service_locator module, 211

tests.blackfennec.util.test_meta_info module, 211

tests.blackfennec.util.test_observable module, 211

tests.context module, 214

tests.test_utils module, 214

tests.test_utils.connection module, 212

tests.test_utils.deep_compare module, 212

tests.test_utils.observer module, 214

tests.test_utils.parameterize module, 214

tests.test_utils.what_the_farmer_does_not_eat_visitor module, 214

text (blackfennec.presentation_system.ui_service.message.Message property), 128

timeout (blackfennec.presentation_system.ui_service.message.Message property), 128

try_determine_mime_type() (blackfennec.document_system.mime_type.mime_type.MimeType static method), 108

try_determine_resource_type() (blackfennec.document_system.resource_type.resource_type.ResourceType static method), 112

type (blackfennec.action_system.action.Action attribute), 104

TYPE (blackfennec.structure.reference.Reference attribute), 135

type (blackfennec.type_system.interpretation.offer.Offer property), 140

Type (class in blackfennec.type_system.type), 143

type() (in module tests.blackfennec.type_system.interpretation.test_offer), 204

type() (in module tests.blackfennec.type_system.test_integration_service_locator), 206

type() (in module tests.blackfennec.type_system.test_null_type), 207

type() (in module tests.blackfennec.type_system.test_type_base), 209

type() (tests.blackfennec.type_system.test_boolean_type.TestBooleanType method), 205

type() (tests.blackfennec.type_system.test_number_type.TestNumberType method), 208

type_loader (blackfennec.extension_system.extension_api.ExtensionApi property), 116

type_loader() (in module tests.blackfennec.extension_system.test_extension_api), 179

type_registry (blackfennec.extension_system.extension_api.ExtensionApi property), 116

type_registry() (in module tests.blackfennec.extension_system.test_extension_api), 179

type_registry() (in module tests.blackfennec.type_system.interpretation.test_interpretation_service_locator), 204

type_registry() (in module tests.blackfennec.type_system.test_integration_type_loader), 206

TypeComparator (class in blackfennec.util.type_comparator), 148

TypeCoverageMixin (class in blackfennec.type_system.type_coverage_mixin), 144

TypeFactory (class in blackfennec.type_system.type_factory), 144

TypeLoader (class in blackfennec.type_system.type_loader), 145

TypeLoaderMock (class in blackfennec_doubles.type_system.double_type_loader), 168

TypeMock (class in blackfennec_doubles.type_system.double_type), 167

TypeParser (class in blackfennec.type_system.type_parser), 145

TypeParserMock (class in blackfennec_doubles.type_system.double_type_parser), 168

TypeRegistry (class in blackfennec.type_system.type_registry), 145

TypeRegistryMock (class in blackfennec_doubles.type_system.double_type_registry), 168

TypeRegistryTestSuite (class in blackfennec.type_system.test_type_registry), 209

types (blackfennec.type_system.interpretation.interpretation.Interpretation property), 138

types (blackfennec.type_system.type_registry.TypeRegistry property), 145

types (blackfennec_doubles.type_system.double_type_registry.TypeRegistry property), 168

types() (in module tests.blackfennec.type_system.interpretation.test_integration_service_locator), 203

TypeView (class in blackfennec.presentation_system.type_view), 130

TypeViewFactory (class in blackfennec.presentation_system.type_view), 130

nec.presentation_system.type_view_factory), 130

TypeViewFactoryRegistry (class in *blackfennec.presentation_system.type_view_factory_registry*), 130

U

ui_service (blackfennec.extension_system.extension_api.ExtensionApi property), 116

ui_service() (in module *tests.blackfennec.extension_system.test_extension_api*), 179

ui_service() (in module *tests.blackfennec.presentation_system.ui_service.test_ui_service*), 195

UiContextMock (class in *blackfennec_doubles.action_system.double_ui_context*), 150

UiService (class in *blackfennec.presentation_system.ui_service.ui_service*), 128

UiServiceMock (class in *blackfennec_doubles.presentation_system.ui_service.double_ui_service*), 160

UiServiceRegistryMock (class in *blackfennec_doubles.presentation_system.ui_service.double_ui_service_registry*), 160

UNDO (*tests.blackfennec.layers.observable.test_integration.observable* attribute), 188

UNDO (*tests.blackfennec.presentation_system.test_history_service.history_service* attribute), 196

undo() (*blackfennec.presentation_system.history_service.HistoryService* method), 129

undo() (*blackfennec.presentation_system.main_window.blackfennec.presentation_system.test_view_factory* method), 125

undo() (*blackfennec_doubles.presentation_system.double_ui_service_registry.view_factory* method), 161

uri (*blackfennec.document_system.document.Document* property), 113

uri (*blackfennec.presentation_system.main_window.document_tab.DocumentTab* property), 126

uri_navigator() (in module *tests.blackfennec.structure.reference_navigation.test_uri_navigation*), 199

UriNavigator (class in *blackfennec.structure.reference_navigation.uri_navigator*), 133

V

validate() (blackfennec.type_system.string_type.StringType method), 143

ValidExtensionMock (class in *blackfennec_doubles.extension_system.double_extensions.valid_extension*), 155

value (*blackfennec.layers.encapsulation_base.encapsulation_base.EncapsulationBase* property), 117

value (*blackfennec.layers.encapsulation_base.list_encapsulation_base.ListEncapsulationBase* property), 118

value (*blackfennec.layers.encapsulation_base.map_encapsulation_base.MapEncapsulationBase* property), 119

value (*blackfennec.layers.merge.merged_list.MergedList* property), 120

value (*blackfennec.layers.merge.merged_map.MergedMap* property), 120

value (*blackfennec.layers.merge.merged_null.MergedNull* property), 120

value (*blackfennec.layers.merge.merged_phantom.MergedPhantom* property), 120

value (*blackfennec.layers.merge.merged_structure.MergedStructure* property), 121

value (*blackfennec.structure.list.List* property), 134

value (*blackfennec.structure.map.Map* property), 134

value (*blackfennec.structure.null.Null* property), 135

value (*blackfennec.structure.reference.Reference* property), 135

value (*blackfennec.structure.structure.Structure* property), 136

value (*blackfennec.structure.structure.ValueStructure* property), 136

value (*blackfennec_doubles.structure.double_structure.NotifyingStructure* property), 165

value (*blackfennec_doubles.structure.double_structure.StructureMock* property), 165

ValueStructure (class in *blackfennec.structure.structure*), 136

view_factory (in module *tests.BlackFennec.ViewMode*), 180

view_factory (*blackfennec.extension_system.test_view_factory*), 180

view_factory (*blackfennec.extension_system.test_view_factory*), 180

view_factory (*blackfennec.extension_system.test_view_factory*), 180

view_factory_registry (blackfennec.extension_system.extension_api.ExtensionApi property), 116

view_factory_registry() (in module *tests.blackfennec.extension_system.test_view_factory*), 179

view_model() (in module method), 117
tests.blackfennec.presentation_system.main_window.visit_list() (blackfennec.layers.merge.merger.ListMerger method), 192

ViewFactoryMock (class in blackfennec_doubles.presentation_system.double_view_factory), 162

ViewFactoryRegistryMock (class in blackfennec_doubles.presentation_system.double_view_factory), 162

visit_boolean() (blackfennec.layers.encapsulation_base.base_factory_visitor.visit_boolean() method), 117

visit_boolean() (blackfennec.layers.merge.merger.BooleanMerger method), 121

visit_boolean() (blackfennec.layers.merge.merger.MergerFactory method), 121

visit_boolean() (blackfennec.layers.merge.merger.NullMerger method), 122

visit_boolean() (blackfennec.layers.merge.merger.Visitor.visit_boolean() method), 137

visit_boolean() (blackfennec.layers.merge.merger.BooleanMerger method), 121

visit_boolean() (blackfennec.layers.merge.merger.MergerFactory method), 121

visit_boolean() (blackfennec.structure.visitor.Visitor.visit_boolean() method), 137

visit_boolean() (blackfennec.type_system.boolean_type.BooleanType method), 141

visit_boolean() (blackfennec.type_system.type_coverage_mixin.TypeCoverageMixin method), 144

visit_boolean() (blackfennec.util.deep_copy.DeepCopyVisitor method), 146

visit_boolean() (blackfennec.util.intercepting_visitor.InterceptingVisitor method), 147

visit_boolean() (blackfennec.type_system.type_coverage_mixin.TypeCoverageMixin method), 144

visit_boolean() (blackfennec.util.deep_copy.DeepCopyVisitor method), 146

visit_boolean() (blackfennec.util.intercepting_visitor.InterceptingVisitor method), 147

visit_boolean() (blackfennec.util.parameterized_visitor.ParameterizedVisitor method), 148

visit_boolean() (blackfennec.util.type_comparator.TypeComparator method), 148

visit_boolean() (blackfennec.util.intercepting_visitor.InterceptingVisitor method), 147

visit_boolean() (blackfennec.util.parameterized_visitor.ParameterizedVisitor method), 148

visit_boolean() (blackfennec.util.type_comparator.TypeComparator method), 148

visit_boolean() (blackfennec_doubles.layers.encapsulation_base.double_factory_base_visitor.visit_boolean() method), 158

visit_boolean() (blackfennec.util.parameterized_visitor.ParameterizedVisitor method), 148

visit_boolean() (blackfennec.util.type_comparator.TypeComparator method), 148

visit_boolean() (blackfennec_doubles.layers.encapsulation_base.double_factory_base_visitor.visit_boolean() method), 158

visit_boolean() (blackfennec_doubles.type_system.double_type_parser.TypeParserMock method), 168

visit_boolean() (tests.test_utils.deep_compare.BooleanComparator method), 212

visit_boolean() (tests.test_utils.deep_compare.ComparatorFactory method), 212

visit_boolean() (tests.test_utils.deep_compare.ComparatorTemplate method), 212

visit_boolean() (tests.test_utils.what_the_farmer_does_not_eat_visitor_factory.visit_boolean() method), 214

visit_boolean() (tests.test_utils.deep_compare.BooleanComparator method), 212

visit_boolean() (tests.test_utils.deep_compare.ComparatorFactory method), 212

visit_boolean() (tests.test_utils.deep_compare.ComparatorTemplate method), 212

visit_boolean() (tests.test_utils.what_the_farmer_does_not_eat_visitor_factory.visit_boolean() method), 214

visit_list() (blackfennec.layers.encapsulation_base.base_factory_visitor.visit_list() method), 121

nec.layers.merge.merger.NullMerger method), 122
visit_map() (*blackfennec.structure.visitor.Visitor* method), 137
visit_map() (*blackfennec.type_system.map_type.MapType* method), 142
visit_map() (*blackfennec.type_system.type_coverage_mixin.TypeCoverageMixin* method), 144
visit_map() (*blackfennec.util.deep_copy.DeepCopyVisitor* method), 146
visit_map() (*blackfennec.util.intercepting_visitor.InterceptingVisitor* method), 147
visit_map() (*blackfennec.util.parameterized_visitor.ParameterizedVisitor* method), 148
visit_map() (*blackfennec.util.type_comparator.TypeComparator* method), 148
visit_map() (*blackfennec_doubles.layers.encapsulation_base.double_factory_base_visitor.FactoryBaseVisitorMock* method), 158
visit_map() (*blackfennec_doubles.type_system.double_type_parser.TypeParserMock* method), 168
visit_map() (*tests.test_utils.deep_compare.ComparatorFactory* method), 212
visit_map() (*tests.test_utils.deep_compare.ComparatorTemplate* method), 212
visit_map() (*tests.test_utils.deep_compare.MapComparator* method), 213
visit_map() (*tests.test_utils.what_the_farmer_does_not_eat_visitor_factory.WhatTheFarmerDoesNotEatVisitorFactory* method), 214
visit_null() (*blackfennec.layers.encapsulation_base.base_factory_visitor.BaseFactoryVisitor* method), 117
visit_null() (*blackfennec.layers.merge.merger.BooleanMerger* method), 121
visit_null() (*blackfennec.layers.merge.merger.ListMerger* method), 121
visit_null() (*blackfennec.layers.merge.merger.MapMerger* method), 121
visit_null() (*blackfennec.layers.merge.merger.Merger* method), 121
visit_null() (*blackfennec.layers.merge.merger.MergerFactory* method), 121
visit_null() (*blackfennec.layers.merge.merger.NullMerger* method), 122
visit_null() (*blackfennec.layers.merge.merger.NumberMerger* method), 122
visit_null() (*blackfennec.layers.merge.merger.StringMerger* method), 122
visit_null() (*blackfennec.structure.visitor.Visitor* method), 137
visit_null() (*blackfennec.type_system.null_type.NullType* method), 142
visit_null() (*blackfennec.type_system.type_coverage_mixin.TypeCoverageMixin* method), 144
visit_null() (*blackfennec.util.deep_copy.DeepCopyVisitor* method), 146
visit_null() (*blackfennec.util.intercepting_visitor.InterceptingVisitor* method), 147
visit_null() (*blackfennec.util.parameterized_visitor.ParameterizedVisitor* method), 148
visit_null() (*blackfennec.util.type_comparator.TypeComparator* method), 148
visit_null() (*blackfennec_doubles.layers.encapsulation_base.double_factory_base_visitor.FactoryBaseVisitorMock* method), 158
visit_null() (*blackfennec_doubles.type_system.double_type_parser.TypeParserMock* method), 168
visit_null() (*tests.test_utils.deep_compare.ComparatorFactory* method), 212
visit_null() (*tests.test_utils.deep_compare.ComparatorTemplate* method), 212
visit_null() (*tests.test_utils.deep_compare.MapComparator* method), 213
visit_null() (*tests.test_utils.what_the_farmer_does_not_eat_visitor_factory.WhatTheFarmerDoesNotEatVisitorFactory* method), 214
visit_number() (*blackfennec.layers.encapsulation_base.base_factory_visitor.BaseFactoryVisitor* method), 117
visit_number() (*blackfennec.layers.merge.merger.MergerFactory* method), 121
visit_number() (*blackfennec.layers.merge.merger.NumberMerger* method), 122
visit_number() (*blackfennec.structure.visitor.Visitor* method), 137

visit_number()	(blackfennec.type_system.number_type.NumberType method), 142	visit_reference()	(blackfennec.util.parameterized_visitor.ParameterizedVisitor method), 148
visit_number()	(blackfennec.type_system.type_coverage_mixin.TypeCoverageMixin method), 144	visit_reference()	(blackfennec.util.type_comparator.TypeComparator method), 149
visit_number()	(blackfennec.util.deep_copy.DeepCopyVisitor method), 146	visit_reference()	(blackfennec_doubles.layers.encapsulation_base.double_factory_base_visitor method), 158
visit_number()	(blackfennec.util.intercepting_visitor.InterceptingVisitor method), 147	visit_reference()	(blackfennec_doubles.type_system.double_type_parser.TypeParserMock method), 168
visit_number()	(blackfennec.util.parameterized_visitor.ParameterizedVisitor method), 148	visit_reference()	(tests.test_utils.deep_compare.ComparatorFactory method), 212
visit_number()	(blackfennec.util.type_comparator.TypeComparator method), 148	visit_reference()	(tests.test_utils.deep_compare.ComparatorTemplate method), 213
visit_number()	(blackfennec_doubles.layers.encapsulation_base.double_factory_base_visitor method), 158	visit_reference()	(tests.test_utils.deep_compare.ReferenceComparator method), 213
visit_number()	(blackfennec_doubles.type_system.double_type_parser.TypeParserMock method), 168	visit_reference()	(tests.test_utils.deep_compare.ReferenceComparator method), 214
visit_number()	(tests.test_utils.deep_compare.ComparatorFactory method), 212	visit_root()	(blackfennec_doubles.type_system.double_type_parser.TypeParserMock method), 168
visit_number()	(tests.test_utils.deep_compare.ComparatorTemplate method), 213	visit_string()	(blackfennec.layers.encapsulation_base.base_factory_visitor.BaseFactoryVisitor method), 117
visit_number()	(tests.test_utils.deep_compare.NumberComparator method), 213	visit_string()	(blackfennec.layers.merge.merger.MergerFactory method), 122
visit_number()	(tests.test_utils.what_the_farmer_does_not_eat_visitor.WhatTheFarmerDoesNotEatVisitor method), 214	visit_string()	(tests.test_utils.what_the_farmer_does_not_eat_visitor.WhatTheFarmerDoesNotEatVisitor method), 214
visit_reference()	(blackfennec.layers.encapsulation_base.base_factory_visitor.BaseFactoryVisitor method), 117	visit_string()	(blackfennec.layers.merge.merger.StringMerger method), 122
visit_reference()	(blackfennec.layers.merge.merger.MergerFactory method), 121	visit_string()	(blackfennec.structure.visitor.Visitor method), 137
visit_reference()	(blackfennec.layers.overlay.overlay_factory_visitor.OverlayFactoryVisitor method), 123	visit_string()	(blackfennec.type_system.string_type.StringType method), 143
visit_reference()	(blackfennec.structure.visitor.Visitor method), 137	visit_string()	(blackfennec.type_system.type_coverage_mixin.TypeCoverageMixin method), 144
visit_reference()	(blackfennec.type_system.reference_type.ReferenceType method), 143	visit_string()	(blackfennec.util.deep_copy.DeepCopyVisitor method), 147
visit_reference()	(blackfennec.type_system.type_coverage_mixin.TypeCoverageMixin method), 144	visit_string()	(blackfennec.util.intercepting_visitor.InterceptingVisitor method), 147
visit_reference()	(blackfennec.util.deep_copy.DeepCopyVisitor method), 146	visit_string()	(blackfennec.util.parameterized_visitor.ParameterizedVisitor method), 148
visit_reference()	(blackfennec.util.intercepting_visitor.InterceptingVisitor method), 149	visit_string()	(blackfennec.util.type_comparator.TypeComparator method), 149
		visit_string()	(blackfennec.util.type_comparator.TypeComparator method), 149

nec_doubles.layers.encapsulation_base.double_factory_base_visitor.FailToVisitVisitorMock does not eat visitor factory),
 method), 158 214

visit_string() (*blackfennec_doubles.type_system.double_type_parser.TypeParserMock*
 method), 168

visit_string() (*tests.test_utils.deep_compare.ComparatorFactory*
 method), 212

visit_string() (*tests.test_utils.deep_compare.ComparatorTemplate*
 method), 213

visit_string() (*tests.test_utils.deep_compare.StringComparator*
 method), 213

visit_string() (*tests.test_utils.what_the_farmer_does_not_eat_visitor_factory.WhatTheFarmerDoesNotEatVisitorFactory*
 method), 214

visit_structure() (*blackfennec.layers.encapsulation_base.base_factory_visitor.BaseFactoryVisitor*
 method), 117

visit_structure() (*blackfennec.layers.merge.merger.Merger* method),
 121

visit_structure() (*blackfennec.layers.merge.merger.NullMerger* method),
 122

visit_structure() (*blackfennec.structure.visitor.Visitor* method), 137

visit_structure() (*blackfennec.util.deep_copy.DeepCopyVisitor* method),
 147

visit_structure() (*blackfennec.util.intercepting_visitor.InterceptingVisitor*
 method), 147

visit_structure() (*blackfennec.util.parameterized_visitor.ParameterizedVisitor*
 method), 148

visit_structure() (*blackfennec.util.type_comparator.TypeComparator*
 method), 149

visit_structure() (*blackfennec_doubles.type_system.double_type_parser.TypeParserMock*
 method), 168

visit_structure() (*tests.test_utils.deep_compare.ComparatorFactory*
 method), 212

visit_structure() (*tests.test_utils.deep_compare.ComparatorTemplate*
 method), 213

visit_structure() (*tests.test_utils.deep_compare.StructureComparator*
 method), 213

visit_structure() (*tests.test_utils.what_the_farmer_does_not_eat_visitor_factory.WhatTheFarmerDoesNotEatVisitorFactory*
 method), 214

Visitor (class in *blackfennec.structure.visitor*), 137

visitor() (in module
tests.blackfennec.layers.observable.test_observable_factory_visitor),
 190

W

WhatTheFarmerDoesNotEatVisitorFactory (class in