Augeias Documentation

Release 0.2.0

Karel Dieussaert

Oct 01, 2020

Contents

1 Installation **3**

2 Services **5**

3 API Documentation **15**
- 3.1 Collections . . . 15
- 3.2 Stores . . . 15
- 3.3 Uri . . . 20
- 3.4 Views . . . 21

4 History **23**
- 4.1 0.6.0 (09-06-2020) . . . 23
- 4.2 0.5.0 (28-11-2018) . . . 23
- 4.3 0.4.1 (26-10-2017) . . . 23
- 4.4 0.4.0 (24-10-2017) . . . 23
- 4.5 0.3.1 (12-10-2017) . . . 23
- 4.6 0.3.0 (22-08-2017) . . . 24
- 4.7 0.2.0 (07-11-2015) . . . 24
- 4.8 0.1.0 (20-10-2015) . . . 24

5 Glossary **25**

6 Indices and tables **27**

HTTP Routing Table **29**

Python Module Index **31**

Index **33**

Augeias is a small, RESTful, webapplication that allows you to store digital objects in an object store. While it allows you to communicate with you objects as a service, it also decouples the storage implementation from the interface.

For this purpose, Augeias, allows you to define multiple collections. Every collection can be served by a different storage module. So, one collection might store it's objects on the filesystem while another one might store it's objects in an Amazon S3 environment. The client that communicates with Augeias doesn't need to know where the digit alobjects it's receiving are stored.

CHAPTER 1

Installation

Getting an installation of Augeias running is very simple. As always we recommend working with a separate virtualenvironment for this.

```
$ mkvirtualenv augeias_demo
$ pip install -U augeias
```

Once Augeias is installed, you can call upon a pyramid scaffold to generate the demo site.

```
$ pcreate -s augeias augeias_demo
$ cd augeias_demo
```

Just a few more steps and you've got a demo version running.

```
$ pip install -r requirements-dev.txt
$ python setup.py develop
$ pserve development.ini
```

The Augeias demo instance is now running on your localhost at port 6543. To reach it, open your browser and surf to the address http://localhost:6543.

This basic version comes with one configured collection *default*.

CHAPTER 2

Services

The augeias provides services on *Collections*, *Containers* and *Objects*. You can think of them as a three-leveled hierarchy. At the top there are the collections. Each collection contains containers. And within each container are 1 or more objects.

GET /collections

Show a list of all collections. Every collection is a set of containers. They might have some sort of meaning as defined by the implementer. In practice quite often there will be a collection for every application.

Example request

```
GET /collections HTTP/1.1
Host: augeias.onroerenderfgoed.be
Accept: application/json
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json

[
    {
        'collection_key': 'default',
        'uri': 'https://storage.onroerenderfgoed.be/collections/default'
    } , {
        'collection_key': 'my_collection',
        'uri': 'https://storage.onroerenderfgoed.be/collections/my_collection'
    }
]
```

Request Headers

- Accept – The response content type depends on this header. Currently only *application/json* is supported.

Response Headers

- Content-Type – This service currently always returns *application/json*

POST /collections/{collection_key}/containers

Create a new container. The server will generate a random container key.

Example request:

```
POST /collections/mine/containers HTTP/1.1
Host: augeias.onroerenderfgoed.be
Accept: application/json
```

Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: https://storage.onroerenderfgoed.be/collections/mine/containers/
↪6ed5a007-41cf-49ed-8cb8-184fa5f48e42

{
    'container_key': '6ed5a007-41cf-49ed-8cb8-184fa5f48e42',
    'uri': 'https://storage.onroerenderfgoed.be/collections/mine/containers/
↪6ed5a007-41cf-49ed-8cb8-184fa5f48e42'
}
```

Parameters

- **collection_key** – Key for the collection within which the container will be placed.

Request Headers

- Accept – The response content type depends on this header. Currently only *application/json* is supported.

Response Headers

- Location – The url where the newly added container can be found.
- Content-Type – This service currently always returns *application/json*

Status Codes

- 201 Created – The container was added succesfully.
- 404 Not Found – The collection *collection_key* does not exist.

PUT /collections/{collection_key}/containers/{container_key}

Create a new container. Allowsyou to choose your own container key. If the container already exists, nothing will happen.

Example request:

```
PUT /collections/mine/containers/abcd HTTP/1.1
Host: augeias.onroerenderfgoed.be
Accept: application/json
```

Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: https://storage.onroerenderfgoed.be/collections/mine/containers/abcd
```

(continues on next page)

(continued from previous page)

```
{
    'container_key': 'abcd'
    'uri': 'https://storage.onroerenderfgoed.be/collections/mine/containers/abcd'
}
```

Parameters

- **collection_key** – Key for the collection within which the container will be placed.
- **container_key** – Key for the container that will be created.

Request Headers

- Accept – The response content type depends on this header. Currently only *application/json* is supported.

Response Headers

- Location – The url where the newly added container can be found.
- Content-Type – This service currently always returns *application/json*

Status Codes

- 200 OK – The container already existed.
- 201 Created – The container was added succesfully.
- 404 Not Found – The collection *collection_key* does not exist.

DELETE /collections/{collection_key}/containers/{container_key}

Remove a container and all the objects in it.

Example request:

```
DELETE /collections/mine/containers/abcd HTTP/1.1
Host: augeias.onroerenderfgoed.be
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    'container_key': 'abcd'
    'uri': 'https://storage.onroerenderfgoed.be/collections/mine/containers/abcd'
}
```

Parameters

- **collection_key** – Key for the collection where the container lives.
- **container_key** – Key for the container that will be deleted.

Request Headers

- Accept – The response content type depends on this header. Currently only *application/json* is supported.

Response Headers

- Content-Type – This service currently always returns *application/json*

Status Codes

- 200 OK – The container was deleted.
- 404 Not Found – The collection *collection_key* does not exist or the container *container_key* does not exist within this collection.

GET /collections/{collection_key}/containers/{container_key}

Show all objects present in this container.

Example request:

```
GET /collections/mine/containers/a311efb7-f125-4d0a-aa26-69d3657a2d06 HTTP/1.1
Host: augeias.onroerenderfgoed.be
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
    "square",
    "small",
    "medium",
    "original",
    "large",
    "full"
]
```

Parameters

- **collection_key** – Key for the collection where the container lives.
- **container_key** – Key for the container that will be queried.

Request Headers

- Accept – The response content type depends on this header. Currently only *application/json* is supported.

Response Headers

- Content-Type – This service currently always returns *application/json*

Status Codes

- 200 OK – The container exists.
- 404 Not Found – The collection *collection_key* or the container *container_key* does not exist.

HEAD /collections/{collection_key}/containers/{container_key}/{object_key}

Fetch metadata on a object without actually fetching the object.

Example request:

```
HEAD /collections/mine/containers/a311efb7-f125-4d0a-aa26-69d3657a2d06/full HTTP/
↪1.1
Host: augeias.onroerenderfgoed.be
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Date: Fri, 30 Oct 2015 07:11:44 GMT
Server: Apache/2.4.7 (Ubuntu)
Content-type: image/jpeg
Content-Length: 23562
```

Parameters

- **collection_key** – Key for the collection where the container lives.
- **container_key** – Key for the container where the object lives.
- **object_key** – Key for the object that will be fetched

Status Codes

- 200 OK – The object was found.
- 404 Not Found – The collection *collection_key* or the container *container_key* or the *object_key* does not exist.

GET /collections/{collection_key}/containers/{container_key}/{object_key}

Fetch an object from a container.

Example request:

```
GET /collections/mine/containers/a311efb7-f125-4d0a-aa26-69d3657a2d06/full HTTP/1.
↪1
Host: augeias.onroerenderfgoed.be
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Date: Fri, 30 Oct 2015 07:11:44 GMT
Server: Apache/2.4.7 (Ubuntu)
Content-type: image/jpeg
Content-Length: 23562

<snipped>
```

Parameters

- **collection_key** – Key for the collection where the container lives.
- **container_key** – Key for the container where the object lives.
- **object_key** – Key for the object that will be fetched

Status Codes

- 200 OK – The object was found.
- 404 Not Found – The collection *collection_key* or the container *container_key* or the *object_key* does not exist.

GET /collections/{collection_key}/containers/{container_key}/{object_key}/meta

Fetch object info (mimetype, size, time last modification).

Example request:

```
GET /collections/mine/containers/a311efb7-f125-4d0a-aa26-69d3657a2d06/full/meta␣
↪HTTP/1.1
Host: augeias.onroerenderfgoed.be
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Content-type: image/jpeg

{
        "time_last_modification": "2017-08-18T13:52:25.970242",
        "mime": "image/jpeg",
        "size": 11370
}
```

Parameters

- **collection_key** – Key for the collection where the container lives.
- **container_key** – Key for the container where the object lives.
- **object_key** – Key for the object that will be fetched

Status Codes

- 200 OK – The object was found.
- 404 Not Found – The collection *collection_key* or the container *container_key* or the *object_key* does not exist.

POST /collections/{collection_key}/containers/{container_key}

Create a new object. The server will generate a random object key.

Example request:

```
POST /collections/mine/containers/mine_container HTTP/1.1
Host: augeias.onroerenderfgoed.be
Content-Type: application/octet-stream
Accept: application/json
```

Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: https://storage.onroerenderfgoed.be/collections/mine/containers/mine_
↪container/6ed5a007-41cf-49ed-8cb8-184fa5f48e42

{
    'container_key': 'mine_container',
    'object_key': '6ed5a007-41cf-49ed-8cb8-184fa5f48e42',
    'collection_key': 'mine'
    'uri': 'https://storage.onroerenderfgoed.be/collections/mine/containers/mine_
↪container/6ed5a007-41cf-49ed-8cb8-184fa5f48e42'
}
```

Parameters

- **collection_key** – Key for the collection where the container lives.

- **container_key** – Key for the container where the object lives.
- **object_key** – Key for the object that will be created or updated.

Request Headers

- Content-Type – *application/json* or *application/octet-stream*
- Accept – The response content type depends on this header. Currently only *application/json* is supported.

Response Headers

- Content-Type – This service currently always returns *application/json*

Status Codes

- 201 Created – The object and the key were created.
- 404 Not Found – The collection *collection_key* or the container *container_key* does not exist.

PUT /collections/{collection_key}/containers/{container_key}/{object_key}

Add or update an object in a container.

If an object with this key already exists, it will be overwritten. If not, it will be created.

Example request:

```
PUT /collections/mine/containers/a311efb7-f125-4d0a-aa26-69d3657a2d06/circle HTTP/
↪1.1
Host: augeias.onroerenderfgoed.be
Content-Type: application/octet-stream
Accept: application/json
```

Exmaple response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    'uri': 'https://id.erfgoed.net/storage/collections/mine/containers/a311efb7-
↪f125-4d0a-aa26-69d3657a2d06/circle',
    'object_key': 'circle',
    'container_key': 'a311efb7-f125-4d0a-aa26-69d3657a2d06',
    'collection_key': 'mine'
}
```

Parameters

- **collection_key** – Key for the collection where the container lives.
- **container_key** – Key for the container where the object lives.
- **object_key** – Key for the object that will be created or updated.

Request Headers

- Content-Type – *application/json* or *application/octet-stream*
- Accept – The response content type depends on this header. Currently only *application/json* is supported.

Response Headers

- Content-Type – This service currently always returns *application/json*

Status Codes

- 200 OK – The object was updated.
- 201 Created – There was no object present with this key, it was created.
- 404 Not Found – The collection *collection_key* or the container *container_key* does not exist.

PUT /collections/{collection_key}/containers/{container_key}/{object_key}

Copy an object from one store location into another within the same Augeias instance. The input json data must contain host url, collection_key, container_key and object_key of the object that needs to be copied.

If an object with this key already exists, it will be overwritten. If not, it will be created.

Example request:

```
PUT /collections/mine/containers/a311efb7-f125-4d0a-aa26-69d3657a2d06/circle HTTP/
↪1.1
Host: augeias.onroerenderfgoed.be
Content-Type: application/json
Accept: application/json

{
  "host_url": "http://augeias.onroerenderfgoed.be",
  "collection_key": "temp",
  "container_key": "container_id",
  "object_key": "circletemp"
}
```

Exmaple response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "uri": "https://id.erfgoed.net/storage/collections/mine/containers/a311efb7-
↪f125-4d0a-aa26-69d3657a2d06/circle",
    "object_key": "circle",
    "container_key": "a311efb7-f125-4d0a-aa26-69d3657a2d06",
    "collection_key": "mine"
}
```

Parameters

- **collection_key** – Key for the collection where the container lives.
- **container_key** – Key for the container where the object lives.
- **object_key** – Key for the object that will be created or updated.

Request Headers

- Content-Type – *application/json* or *application/octet-stream*
- Accept – The response content type depends on this header. Currently only *application/json* is supported.

Response Headers

- Content-Type – This service currently always returns *application/json*

Status Codes

- 200 OK – The object was copied.
- 201 Created – There was no object present with this key, it was created.
- 400 Bad Request – Validation failure. The input data of the object in the json body is not correct.
- 404 Not Found – The collection *collection_key* or the container *container_key* does not exist.

DELETE /collections/{collection_key}/containers/{container_key}/{object_key}

Delete an object from a container.

Example request:

```
DELETE /collections/mine/containers/a311efb7-f125-4d0a-aa26-69d3657a2d06/full␣
↪HTTP/1.1
Host: augeias.onroerenderfgoed.be
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "uri": "https://id.erfgoed.net/storage/collections/mine/containers/a311efb7-
↪f125-4d0a-aa26-69d3657a2d06/circle",
    "object_key":"full",
    "container_key":"a311efb7-f125-4d0a-aa26-69d3657a2d06",
    "collection_key": "mine"
}
```

Status Codes

- 200 OK – The object was deleted.
- 404 Not Found – The collection *collection_key* or the container *container_key* or the *object_key* does not exist.

CHAPTER 3

API Documentation

3.1 Collections

Contains models that deal with collections.

class `augeias.collections.model.`**`Collection`**(*name*, *object_store*, ***kwargs*)

A collection is a set of containers that can have separate logic and configuration applied to it.

Currently a collection allows making two chief distinctions:

- Where data whilebe stored, by setting the *object_store*
- How URI's for the collection and it's contents will be generated by setting a *uri_generator*

3.2 Stores

3.2.1 Errors

This module contains exceptions that can be generated by stores.

exception `augeias.stores.error.`**`NotFoundException`**(*value='Nothing matching the key value was found in the store'*)

Indicates that an object or container could not be found.

Check the error message for more detailed information.

3.2.2 Store Interface

This module defines the interface every store needs to adhere to.

class `augeias.stores.StoreInterface.`**`IStore`**

This interface handles object-storage. Implementations of this interface can be made for different object-storages Currently this interface is only implemented for PairTreeFileSystemStore

create_container (*container_key*)
Create a new container in the data store.

Parameters **container_key** (*str*) – Key of the container to create.

create_object (*container_key*, *object_key*, *object_data*)
Save a new object in the data store

Parameters

- **container_key** (*str*) – Key of the container to create an object in.
- **object_key** (*str*) – Key of the object to create.
- **object_data** (*str*) – The data for the object to create.

Raises *augeias.stores.error.NotFoundException* – When the container could not be found.

delete_container (*container_key*)
Delete a container and all it's objects in the data store.

Parameters **container_key** (*str*) – Key of the container to delete.

Raises *augeias.stores.error.NotFoundException* – When the container could not be found.

delete_object (*container_key*, *object_key*)
Delete an object from the data store.

Parameters

- **container_key** (*str*) – Key of the container that the object lives in.
- **object_key** (*str*) – Key of the object to delete.

Raises *augeias.stores.error.NotFoundException* – When the object or container could not be found.

get_container_data (*container_key*, *translations=None*)
Find a container and return a zip file of its contents.

Parameters

- **container_key** – Key of the container which must be retrieved.
- **translations** – Dict of object IDs and file names to use for them.

Returns a zip file containing all files of the container.

get_object (*container_key*, *object_key*)
Retrieve an object from the data store.

Parameters

- **container_key** (*str*) – Key of the container that the object lives in.
- **object_key** (*str*) – Key of the object to retrieve.

Raises *augeias.stores.error.NotFoundException* – When the object or container could not be found.

get_object_info (*container_key*, *object_key*)
Retrieve object info (mimetype, size, time last modification) from the data store.

Parameters

- **container_key** (*str*) – Key of the container that the object lives in.

- **object_key** (*str*) – Key of the object to retrieve.

Raises *augeias.stores.error.NotFoundException* – When the object or container could not be found.

list_object_keys_for_container(*container_key*)
List all object keys for a container in the data store.

Parameters **container_key** (*str*) – Key of the container to list the objects for.

Returns A list of container keys.

Return type lst

Raises *augeias.stores.error.NotFoundException* – When the container could not be found.

update_object(*container_key*, *object_key*, *object_data*)
Update an object in the data store.

Parameters

- **container_key** (*str*) – Key of the container that the object lives in.
- **object_key** (*str*) – Key of the object to update.
- **object_data** (*str*) – New data for the object.

Raises *augeias.stores.error.NotFoundException* – When the object or container could not be found.

3.2.3 Implementations

This module provide a simple filesystem based store

class augeias.stores.PairTreeFileSystemStore.**PairTreeFileSystemStore**(*store_dir*, *uri_base='urn:x-vioe:'*)

Provides a filesystem based store.

Will store your digital objects on disk using a PairTree.

create_container(*container_key*)
Create a new container in the data store.

Parameters **container_key** (*str*) – Key of the container to create.

create_object(*container_key*, *object_key*, *object_data*)
Save a new object in the data store

Parameters

- **container_key** (*str*) – Key of the container to create an object in.
- **object_key** (*str*) – Key of the object to create.
- **object_data** (*str*) – The data for the object to create.

Raises *augeias.stores.error.NotFoundException* – When the container could not be found.

delete_container(*container_key*)
Delete a container and all it's objects in the data store.

Parameters **container_key** (*str*) – Key of the container to delete.

Raises *augeias.stores.error.NotFoundException* – When the container could not be found.

delete_object (*container_key*, *object_key*)
Delete an object from the data store.

Parameters

- **container_key** (*str*) – Key of the container that the object lives in.
- **object_key** (*str*) – Key of the object to delete.

Raises *augeias.stores.error.NotFoundException* – When the object or container could not be found.

get_container_data (*container_key*, *translations=None*)
Find a container and return a zip file of its contents.

Parameters

- **container_key** – Key of the container which must be retrieved.
- **translations** – Dict of object IDs and file names to use for them.

Returns a zip file containing all files of the container.

get_object (*container_key*, *object_key*)
Retrieve an object from the data store.

Parameters

- **container_key** (*str*) – Key of the container that the object lives in.
- **object_key** (*str*) – Key of the object to retrieve.

Raises *augeias.stores.error.NotFoundException* – When the object or container could not be found.

get_object_info (*container_key*, *object_key*)
Retrieve object info (mimetype, size, time last modification) from the data store.

Parameters

- **container_key** (*str*) – Key of the container that the object lives in.
- **object_key** (*str*) – Key of the object to retrieve.

Raises *augeias.stores.error.NotFoundException* – When the object or container could not be found.

list_object_keys_for_container (*container_key*)
List all object keys for a container in the data store.

Parameters **container_key** (*str*) – Key of the container to list the objects for.

Returns A list of container keys.

Return type lst

Raises *augeias.stores.error.NotFoundException* – When the container could not be found.

update_object (*container_key*, *object_key*, *object_data*)
Update an object in the data store.

Parameters

- **container_key** (*str*) – Key of the container that the object lives in.

- **object_key** (*str*) – Key of the object to update.
- **object_data** (*str*) – New data for the object.

Raises ***augeias.stores.error.NotFoundException*** – When the object or container could not be found.

class augeias.stores.CephStore.**CephStore**

create_container (*container_key*)
Create a new container in the data store.

Parameters **container_key** (*str*) – Key of the container to create.

create_object (*container_key*, *object_key*, *object_data*)
Save a new object in the data store

Parameters

- **container_key** (*str*) – Key of the container to create an object in.
- **object_key** (*str*) – Key of the object to create.
- **object_data** (*str*) – The data for the object to create.

Raises ***augeias.stores.error.NotFoundException*** – When the container could not be found.

delete_container (*container_key*)
Delete a container and all it's objects in the data store.

Parameters **container_key** (*str*) – Key of the container to delete.

Raises ***augeias.stores.error.NotFoundException*** – When the container could not be found.

delete_object (*container_key*, *object_key*)
Delete an object from the data store.

Parameters

- **container_key** (*str*) – Key of the container that the object lives in.
- **object_key** (*str*) – Key of the object to delete.

Raises ***augeias.stores.error.NotFoundException*** – When the object or container could not be found.

get_container_data (*container_key*, *translations=None*)
Find a container and return a zip file of its contents.

Parameters

- **container_key** – Key of the container which must be retrieved.
- **translations** – Dict of object IDs and file names to use for them.

Returns a zip file containing all files of the container.

get_object (*container_key*, *object_key*)
Retrieve an object from the data store.

Parameters

- **container_key** (*str*) – Key of the container that the object lives in.
- **object_key** (*str*) – Key of the object to retrieve.

Raises *augeias.stores.error.NotFoundException* – When the object or container could not be found.

get_object_info(*container_key*, *object_key*)
Retrieve object info (mimetype, size, time last modification) from the data store.

Parameters

- **container_key** (*str*) – Key of the container that the object lives in.
- **object_key** (*str*) – Key of the object to retrieve.

Raises *augeias.stores.error.NotFoundException* – When the object or container could not be found.

list_object_keys_for_container(*container_key*)
List all object keys for a container in the data store.

Parameters **container_key** (*str*) – Key of the container to list the objects for.

Returns A list of container keys.

Return type lst

Raises *augeias.stores.error.NotFoundException* – When the container could not be found.

update_object(*container_key*, *object_key*, *object_data*)
Update an object in the data store.

Parameters

- **container_key** (*str*) – Key of the container that the object lives in.
- **object_key** (*str*) – Key of the object to update.
- **object_data** (*str*) – New data for the object.

Raises *augeias.stores.error.NotFoundException* – When the object or container could not be found.

3.3 Uri

This module provides utilities for working with *URIS*.

class augeias.uri.**DefaultUriGenerator**(*pattern=u'https://storage.onroerenderfgoed.be/'*)
Generate a *URI* specific to storageprovider.

Used for providers that do not implement a specific *UriGenerator*.

generate_collection_uri(*collection*)
Generate a *URI* for collections based on parameters passed.

Parameters **collection** – The collection.

Return type string

generate_container_uri(*collection*, *container*)
Generate a *URI* for collection and container based on parameters passed.

Parameters

- **collection** – The collection.
- **container** – The container.

Return type string

generate_object_uri(*collection*, *container*, *object*)
Generate a *URI* for collections, container and object based on parameters passed.

Parameters

- **collection** – The collection.
- **container** – The container.
- **object** – The object.

Return type string

class augeias.uri.**UriGenerator**
An abstract class for generating URIs.

generate_collection_uri(*collection*)
Generate a *URI* for a collection based on parameters passed.

generate_container_uri(*collection*, *container*)
Generate a *URI* for a container based on parameters passed.

generate_object_uri(*collection*, *container*, *object*)
Generate a *URI* for a object based on parameters passed.

class augeias.uri.**UriPatternGenerator**(*pattern*, *separator=u'/'*)
Generate a *URI* based on a simple pattern.

generate_collection_uri(*collection*)
Generate a *URI* for a collection based on parameters passed.

Parameters **collection** – The collection.

Return type string

generate_container_uri(*collection*, *container*)
Generate a *URI* for collection and container based on parameters passed.

Parameters

- **collection** – The collection.
- **container** – The container.

Return type string

generate_object_uri(*collection*, *container*, *object*)
Generate a *URI* for collections, container and object based on parameters passed.

Parameters

- **collection** – The collection.
- **container** – The container.
- **object** – The object.

Return type string

3.4 Views

exception augeias.views.**ValidationFailure**(*msg*)

CHAPTER 4

History

4.1 0.6.0 (09-06-2020)

- Lijst met vertaling van id naar bestandsnaam meesturen download zip (#80)

4.2 0.5.0 (28-11-2018)

- Download zip of entire container (#48)
- Update list of supported python version (#51)

4.3 0.4.1 (26-10-2017)

- pairtree update to 0.8.1

4.4 0.4.0 (24-10-2017)

- Added python 3 compatibility

4.5 0.3.1 (12-10-2017)

- Fix pairtree dependency==0.7.1-T

4.6 0.3.0 (22-08-2017)

- Add 'copy' functionality to update_object endpoint. In this case the view accepts the (augeias) location of the file object (host url, collection_key, container_key and object_key) and copies it to a new location on the same augeias instance. (#17)
- Add Endpoint to create object and key. (#20)
- Add Endpoint to retrieve meta data (mimetype (#13), size, time last modification) from object.

4.7 0.2.0 (07-11-2015)

- Add a scaffold for generating an Augeias instance. Use by running *pcreate -s augeias*. (#6)
- Update docs with information about making a HEAD request for an object's metdata. (#9)
- Added permission hooks so it's possible for an Augeias instance to add authentication. (#2)
- Removed some OE specific configuration and made it more generic. (#8)
- Added some installation instructions.

4.8 0.1.0 (20-10-2015)

- Initial version
- Has as single, filesystem based, store.

CHAPTER 5

Glossary

URI A *Uniform Resource Identifier*.

URN A URN is a specific form of a *URI*.

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

HTTP Routing Table

/collections

HEAD /collections/{collection_key}/containers/{container_key}/{object_key}, 8
GET /collections, 5
GET /collections/{collection_key}/containers/{container_key}, 8
GET /collections/{collection_key}/containers/{container_key}/{object_key}, 9
GET /collections/{collection_key}/containers/{container_key}/{object_key}/meta, 9
POST /collections/{collection_key}/containers, 6
POST /collections/{collection_key}/containers/{container_key}, 10
PUT /collections/{collection_key}/containers/{container_key}, 6
PUT /collections/{collection_key}/containers/{container_key}/{object_key}, 12
DELETE /collections/{collection_key}/containers/{container_key}, 7
DELETE /collections/{collection_key}/containers/{container_key}/{object_key}, 13

Python Module Index

a

augeias.collections.model, 15
augeias.stores.CephStore, 19
augeias.stores.error, 15
augeias.stores.PairTreeFileSystemStore, 17
augeias.stores.StoreInterface, 15
augeias.uri, 20
augeias.views, 21

Index

A

augeias.collections.model (*module*), 15
augeias.stores.CephStore (*module*), 19
augeias.stores.error (*module*), 15
augeias.stores.PairTreeFileSystemStore (*module*), 17
augeias.stores.StoreInterface (*module*), 15
augeias.uri (*module*), 20
augeias.views (*module*), 21

C

CephStore (*class in augeias.stores.CephStore*), 19
Collection (*class in augeias.collections.model*), 15
create_container() (*augeias.stores.CephStore.CephStore method*), 19
create_container() (*augeias.stores.PairTreeFileSystemStore.PairTreeFileSystemStore method*), 17
create_container() (*augeias.stores.StoreInterface.IStore method*), 15
create_object() (*augeias.stores.CephStore.CephStore method*), 19
create_object() (*augeias.stores.PairTreeFileSystemStore.PairTreeFileSystemStore method*), 17
create_object() (*augeias.stores.StoreInterface.IStore method*), 16

D

DefaultUriGenerator (*class in augeias.uri*), 20
delete_container() (*augeias.stores.CephStore.CephStore method*), 19
delete_container() (*augeias.stores.PairTreeFileSystemStore.PairTreeFileSystemStore method*), 17
delete_container() (*augeias.stores.StoreInterface.IStore method*), 16
delete_object() (*augeias.stores.CephStore.CephStore method*), 19
delete_object() (*augeias.stores.PairTreeFileSystemStore.PairTreeFil method*), 18
delete_object() (*augeias.stores.StoreInterface.IStore method*), 16

G

generate_collection_uri() (*augeias.uri.DefaultUriGenerator method*), 20
generate_collection_uri() (*augeias.uri.UriGenerator method*), 21
generate_collection_uri() (*augeias.uri.UriPatternGenerator method*), 21
generate_container_uri() (*augeias.uri.DefaultUriGenerator method*), 20
generate_container_uri() (*augeias.uri.UriGenerator method*), 21
generate_container_uri() (*augeias.uri.UriPatternGenerator method*), 21
generate_object_uri() (*augeias.uri.DefaultUriGenerator method*), 21
generate_object_uri() (*augeias.uri.UriGenerator method*), 21
generate_object_uri() (*augeias.uri.UriPatternGenerator method*), 21
get_container_data() (*augeias.stores.CephStore.CephStore method*), 19
get_container_data() (*augeias.stores.PairTreeFileSystemStore.PairTreeFileSystemStore method*), 18
get_container_data() (*augeias.stores.StoreInterface.IStore method*), 16

get_object() (*augeias.stores.CephStore.CephStore method*), 19
get_object() (*augeias.stores.PairTreeFileSystemStore.PairTreeFileSystemStore method*), 18
get_object() (*augeias.stores.StoreInterface.IStore method*), 16
get_object_info() (*augeias.stores.CephStore.CephStore method*), 20
get_object_info() (*augeias.stores.PairTreeFileSystemStore.PairTreeFileSystemStore method*), 18
get_object_info() (*augeias.stores.StoreInterface.IStore method*), 16

I

IStore (*class in augeias.stores.StoreInterface*), 15

L

list_object_keys_for_container() (*augeias.stores.CephStore.CephStore method*), 20
list_object_keys_for_container() (*augeias.stores.PairTreeFileSystemStore.PairTreeFileSystemStore method*), 18
list_object_keys_for_container() (*augeias.stores.StoreInterface.IStore method*), 17

N

NotFoundException, 15

P

PairTreeFileSystemStore (*class in augeias.stores.PairTreeFileSystemStore*), 17

U

update_object() (*augeias.stores.CephStore.CephStore method*), 20
update_object() (*augeias.stores.PairTreeFileSystemStore.PairTreeFileSystemStore method*), 18
update_object() (*augeias.stores.StoreInterface.IStore method*), 17
URI, **25**
UriGenerator (*class in augeias.uri*), 21
UriPatternGenerator (*class in augeias.uri*), 21
URN, **25**

V

ValidationFailure, 21