
MyCapytains Documentation

Release 0.0.1

Thibault Clérice

December 06, 2016

1	MyCapytain.common	3
1.1	MyCapytain.common.metadata	3
1.2	MyCapytain.common.reference	6
1.3	MyCapytain.common.utils	11
2	MyCapytain.resources package	13
2.1	MyCapytain.resources.texts package	13
2.2	MyCapytain.resources.proto.inventory module	16
2.3	MyCapytain.resources.proto.text module	17
3	MyCapytain.endpoints package	21
3.1	MyCapytain.endpoints.ahab module	21
3.2	MyCapytain.endpoints.cts5 module	21
3.3	MyCapytain.endpoints.proto module	22
4	Indices and tables	25
	Python Module Index	27

Contents:

MyCapytain.common

The common namespace aims to regroup together common tools for different parts of the abstraction

1.1 MyCapytain.common.metadata

class `MyCapytain.common.metadata.Metadata` (*keys=None*)

Bases: `future.types.newobject.newobject`

A metadatum aggregation object provided to centralize metadata

Parameters `key` (*List.<basestring>*) – A metadata field name

Variables `metadata` – Dictionary of metadatum

`__getitem__` (*key*)

Add a quick access system through `getitem` on the instance

Parameters `key` (*basestring, int, tuple*) – Index key representing a set of metadatum

Returns An element of children whose index is key

Raises `KeyError` If key is not registered or recognized

Example

```
>>> a = Metadata()
>>> m1 = Metadatum(name="title", [("lat", "Amores"), ("fre", "Les Amours")])
>>> m2 = Metadatum(name="author", [("lat", "Ovidius"), ("fre", "Ovide")])
>>> a[("title", "author")] = (m1, m2)
```

```
>>> a["title"] == m1
>>> a[0] == m1
>>> a[("title", "author")] == (m1, m2)
```

`__setitem__` (*key, value*)

Set a new metadata field

Parameters

- **key** (*basestring, tuple*) – Name of metadatum field
- **value** (*Metadatum*) – Metadatum dictionary

Returns An element of children whose index is key

Raises *TypeError* if key is not basestring or tuple of basestring

Raises *ValueError* if key and value are list and are not the same size

Example

```
>>> a = Metadata()
```

```
>>> a["title"] = Metadatum(name="title", [("lat", "Amores"), ("fre", "Les Amours")])
>>> print(a["title"]["lat"]) # Amores
```

```
>>> a[("title", "author")] = (
>>>     Metadatum(name="title", [("lat", "Amores"), ("fre", "Les Amours")]),
>>>     Metadatum(name="author", [("lat", "Ovidius"), ("fre", "Ovide")])
>>> )
>>> print(a["title"]["lat"], a["author"]["fre"]) # Amores, Ovide
```

__iter__()
Iter method of Metadata

Example

```
>>> a = Metadata(("title", "desc", "author"))
>>> for key, value in a:
>>>     print(key, value) # Print ("title", "<Metadatum object>") then ("desc", "<Metadatum object>") then ("author", "<Metadatum object>")
```

__len__()
Returns the number of Metadatum registered in the object

Return type *int*
Returns Number of metadatum objects

Example

```
>>> a = Metadata(("title", "description", "author"))
>>> print(len(a)) # 3
```

__add__(other)
Merge Metadata objects together

Parameters *other* (*Metadata*) – Metadata object to merge with the current one
Returns The merge result of both metadata object
Return type *Metadata*

Example

```
>>> a = Metadata(name="label")
>>> b = Metadata(name="title")
>>> a + b == Metadata(name=["label", "title"])
```

class MyCapytain.common.metadata.**Metadatum** (*name*, *children=None*)
Bases: *future.types.newobject.newobject*
Metadatum object represent a single field of metadata

Parameters

- **name** (*basestring*) – Name of the field

- **children** (*List*) – List of tuples, where first element is the key, and second the value

Example

```
>>> a = Metadatum(name="label", [("lat", "Amores"), ("fre", "Les Amours")])
>>> print(a["lat"]) # == "Amores"
```

`__getitem__` (*key*)

Add an iterable access method

Int typed key access to the *n* th registered key in the instance. If string based key does not exist, see for a default.

Parameters **key** (*basestring, tuple, int*) – Key of wished value

Returns An element of children whose index is key

Raises `KeyError` if key is unknown (when using Int based key or when default is not set)

Example

```
>>> a = Metadatum(name="label", [("lat", "Amores"), ("fre", "Les Amours")])
>>> print(a["lat"]) # Amores
>>> print(a[("lat", "fre")]) # Amores, Les Amours
>>> print(a[0]) # Amores
>>> print(a["dut"]) # Amores
```

`__setitem__` (*key, value*)

Register index key and value for the instance

Parameters

- **key** (*basestring, list, tuple*) – Index key(s) for the metadata
- **value** (*basestring, list, tuple*) – Values for the metadata

Returns An element of children whose index is key

Raises `TypeError` if key is not basestring or tuple of basestring

Raises `ValueError` if key and value are list and are not the same size

Example

```
>>> a = Metadatum(name="label")
```

```
>>> a["eng"] = "Illiad"
>>> print(a["eng"]) # Illiad
```

```
>>> a[("fre", "grc")] = ("Illiade", "λ")
>>> print(a["fre"], a["grc"]) # Illiade, λ
```

```
>>> a[("ger", "dut")] = "Iliade"
>>> print(a["ger"], a["dut"]) # Iliade, Iliade
```

`__iter__` ()

Iter method of Metadatum

Example

```
>>> a = Metadata(name="label", [("lat", "Amores"), ("fre", "Les Amours")])
>>> for key, value in a:
>>>     print(key, value) # Print ("lat", "Amores") and then ("fre", "Les Amours")
```

setDefault (*key*)

Set a default key when a field does not exist

Parameters *key* (*basestring*) – An existing key of the instance

Returns Default key

Raises *ValueError* If key is not registered

Example

```
>>> a = Metadatum(name="label", [("lat", "Amores"), ("fre", "Les Amours")])
>>> a.setDefault("fre")
>>> print(a["eng"]) # == "Les Amours"
```

1.2 MyCapytain.common.reference

```
>>> from MyCapytain.common.reference import (URN, Reference, Citation)
```

class MyCapytain.common.reference.**Citation** (*name=None, xpath=None, scope=None, refsDecl=None, child=None*)

Bases: `future.types.newobject.newobject`

A citation object gives informations about the scheme

Parameters

- **name** (*basestring*) – Name of the citation (e.g. “book”)
- **xpath** (*basestring*) – Xpath of the citation (As described by CTS norm)
- **scope** – Scope of the citation (As described by CTS norm)
- **refsDecl** (*basestring*) – refsDecl version
- **child** (*Citation*) – A citation

__iter__ ()

Iteration method

Loop over the citation childs

Example

```
>>> c = Citation(name="line")
>>> b = Citation(name="poem", child=c)
>>> a = Citation(name="book", child=b)
>>> [e for e in a] == [a, b, c]
```

__len__ ()

Length method

Return type `int`

Returns Number of nested citations

child

Child of a citation

Type Citation or None

Example Citation.name==poem would have a child Citation.name==line

fill (*passage=None, xpath=None*)

Fill the xpath with given informations

Parameters

- **passage** (*Reference or list*) – Passage reference
- **xpath** (*Boolean*) – If set to True, will return the replaced self.xpath value and not the whole self.refsDecl

Return type basestring

Returns Xpath to find the passage

name

Type of the citation represented

Type basestring

Example Book, Chapter, Textpart, Section, Poem...

refsDecl

RefsDecl expression of the citation scheme

Type basestring

Example /tei:TEI/tei:text/tei:body/tei:div//tei:l[@n='\$1']

scope

TextInventory scope property of a citation (ie. identifier of all element but the last of the citation)

Type basestring

Example /tei:TEI/tei:text/tei:body/tei:div

xpath

TextInventory xpath property of a citation (ie. identifier of the last element of the citation)

Type basestring

Example //tei:l[@n="???"]

MyCapytain.common.reference.**REF_REPLACER** (*match, passage*)

Helper to replace xpath/scope/refsDecl on iteration with passage value

Parameters

- **match** (*re.SRE_MATCH*) – A RegExp match
- **passage** (*iter*) – A list with subreference informations

Return type basestring

Returns Replaced string

class MyCapytain.common.reference.**Reference** (*reference*)

Bases: future.types.newobject.newobject

A reference object giving informations

Parameters **reference** (*basestring*) – Passage Reference part of a Urn

Example

```
>>> a = Reference(reference="1.1@Achiles[1]-1.2@Zeus[1]")
>>> b = Reference(reference="1.1")
```

`__str__()`

Return full reference in string format

Return type `basestring`

Returns String representation of Reference Object

Example

```
>>> a = Reference(reference="1.1@Achiles[1]-1.2@Zeus[1]")
>>> b = Reference(reference="1.1")
>>> str(a) == "1.1@Achiles[1]-1.2@Zeus[1]"
>>> str(b) == "1.1"
```

`__eq__(other)`

Equality checker for Reference object

Parameters `other` – An object to be checked against

Return type `boolean`

Returns Equality between other and self

Example

```
>>> a = Reference(reference="1.1@Achiles[1]-1.2@Zeus[1]")
>>> b = Reference(reference="1.1")
>>> c = Reference(reference="1.1")
>>> (a == b) == False
>>> (c == b) == True
```

`__getitem__(key)`

Return part of or full passage reference

Available keys :

- `1 | start` : First part of the reference
- `2 | start_list` : Reference start parsed into a list
- `3 | start_sub` : Subreference start parsed into a tuple
- `4 | end` : Last part of the reference
- `5 | end_list` : Reference start parsed into a list
- `6 | end_sub` : Subreference end parsed into a tuple
- `default` : full string reference

Parameters `key` (`basestring` or `int`) – Identifier of the part to return

Return type `basestring` or `List.<int>` or `None` or `Tuple.<string>`

Returns Desired part of the passage reference

Example

```

>>> a = Reference(reference="1.1@Achiles[1]-1.2@Zeus[1]")
>>> print(a[1]) # "1.1@Achiles[1]"
>>> print(a["start_list"]) # ("1", "1")
>>> print(a[6]) # ("Zeus", "1")
>>> print(a[7]) # "1.1@Achiles[1]-1.2@Zeus[1]"

```

parent**Returns**

class MyCapytain.common.reference.**URN** (*urn*)
 Bases: future.types.newobject.newobject

A URN object giving all useful sections

Parameters *urn* (*basestring*) – A CTS URN

Example

```
>>> a = URN(urn="urn:cts:latinLit:phi1294.phi002.perseus-lat2:1.1")
```

__len__ ()

Returns the len of the URN

Return type int

Returns Length of the URN

Warning: Does not take into account the passage !

Example

```

>>> a = URN(urn="urn:cts:latinLit:phi1294.phi002.perseus-lat2:1.1")
>>> print(len(a)) #

```

__gt__ (*other*)

Allows for greater comparison

Parameters *other* (*URN*) – Comparison object

Return type boolean

Returns Indicator of bigger size

Warning: Does not take into account the passage !

Example

```

>>> a = URN(urn="urn:cts:latinLit:phi1294.phi002.perseus-lat2:1.1")
>>> b = URN(urn="urn:cts:latinLit:phi1294.phi002:1.1")
>>> (a > b) == True #

```

__lt__ (*other*)

Allows for lower comparison

Parameters *other* (*URN*) – Comparison object

Return type boolean

Returns Indicator of lower size

Warning: Does not take into account the passage !

Example

```
>>> a = URN(urn="urn:cts:latinLit:phi1294.phi002.perseus-lat2:1.1")
>>> b = URN(urn="urn:cts:latinLit:phi1294.phi002:1.1")
>>> (b < a) == True #
```

__eq__ (*other*)

Equality checker for URN object

Parameters *other* (URN) – An object to be checked against

Return type boolean

Returns Equality between other and self

Example

```
>>> a = URN(urn="urn:cts:latinLit:phi1294.phi002.perseus-lat2:1.1")
>>> b = URN(urn="urn:cts:latinLit:phi1294.phi002:1.1")
>>> (b == a) == False #
```

__str__ ()

Return full initial urn

Return type basestring

Returns String representation of URN Object

Example

```
>>> a = URN(urn="urn:cts:latinLit:phi1294.phi002.perseus-lat2:1.1")
>>> str(a) == "urn:cts:latinLit:phi1294.phi002.perseus-lat2:1.1"
```

__getitem__ (*key*)

Returns the urn (int) level or up to (str) level.

Available keys :

- 0 : URN
- *full* : URN
- 1 : Namespace of the urn (cts)
- *urn_namespace* : URN until the Namespace of the urn
- 2 : CTS Namespace of the URN (e.g. latinLit)
- *cts_namespace* : URN until the CTS Namespace
- 3 : Textgroup of the URN
- *textgroup* : URN until the Textgroup
- 4 : Work of the URN
- *work* : URN until the Work
- 5 : Text of the URN
- *text* : URN until the Text
- 6 or *passage*: Passage of URN

Parameters `key` (*int* or *basestring*) – Identifier of the wished resource

Return type *basestring* or *Reference*

Returns Part or complete URN

Warning *urn:* is not counted as an element !

Example

```
>>> a = URN(urn="urn:cts:latinLit:phi1294.phi002.perseus-lat2:1.1")
>>> a["textgroup"] == "urn:cts:latinLit:phi1294"
>>> a[3] == "phi1294"
```

1.3 MyCapytain.common.utils

`MyCapytain.common.utils.NS = {u'xml': u'http://www.w3.org/XML/1998/namespace', u'tei': u'http://www.tei-c.org/ns/1.0'}`
 Dictionary of namespace that can be useful

`MyCapytain.common.utils.normalize` (*string*)

Remove double-or-more spaces in a string

Parameters `string` (*basestring*) – A string to change

Return type *Basestring*

Returns Clean string

`MyCapytain.common.utils.xmlparser` (*xml*)

Parse xml

Parameters `xml` (*basestring*, *lxml.etree._Element*) – XML element

Return type *lxml.etree._Element*

Returns An element object

Raises *TypeError* if element is not in accepted type

MyCapytain.resources package

2.1 MyCapytain.resources.texts package

class `MyCapytain.resources.texts.tei.Citation` (*name=None, xpath=None, scope=None, refs-Decl=None, child=None*)

Bases: `MyCapytain.common.reference.Citation`

Implementation of Citation for TEI markup

static ingest (*resource, xpath='//tei:cRefPattern'*)

Ingest a resource and store data in its instance

Parameters

- **resource** (`lxml.etree._Element`) – XML node `cRefPattern` or list of them in ASC hierarchy order (deepest to highest, eg. lines to poem to book)
- **xpath** (`str`) – XPath to use to retrieve citation

Returns A citation object

Return type `Citation`

class `MyCapytain.resources.texts.tei.Passage` (*parent=None, **kwargs*)

Bases: `MyCapytain.resources.proto.text.Passage`

text (*exclude=None*)

Text content of the passage

Parameters **exclude** (`List`) – Remove some nodes from text

Return type `basestring`

Returns Text of the xml node

Example

```
>>> P = Passage(resource='<l n="8">Ibis <note>hello<a>b</a></note> ab excusso miss
>>> P.text == "Ibis hello b ab excusso missus in astra sago. "
>>> P.text(exclude=["note"]) == "Ibis hello b ab excusso missus in astra sago. "
```

xml

XML Representation of the Passage

Return type `lxml.etree._Element`

Returns XML element representing the passage

class `MyCapytain.resources.texts.local.Text` (*urn=None, citation=None, resource=None*)
Bases: `MyCapytain.resources.proto.text.Text`

Implementation of CTS tools for local files

Parameters

- **urn** (`MyCapytain.common.reference.URN`) – A URN identifier
- **resource** (`lxml.etree._Element`) – A resource
- **citation** (`MyCapytain.common.reference.Citation`) – Highest Citation level

Variables **resource** – lxml

citation

Get the lowest cRefPattern in the hierarchy

Return type `MyCapytain.resources.texts.tei.Citation`

getPassage (*reference*)

Finds a passage in the current text

Parameters **reference** (`List, MyCapytain.common.reference.Reference`) – Identifier of the subreference / passages

Return type `Passage`

Returns Asked passage

getPassagePlus (*reference*)

Finds a passage in the current text with its previous and following node

Parameters **reference** (`List, MyCapytain.common.reference.Reference`) – Identifier of the subreference / passages

Return type `text.PassagePlus`

Returns Asked passage with metainformations

getValidReff (*level=1, reference=None*)

Retrieve valid passages directly

Parameters

- **level** (`Int`) – Depth required. If not set, should retrieve first encountered level (1 based)
- **reference** (`Reference`) – Subreference (optional)

Return type `List.basestring`

Returns List of levels

Note: `GetValidReff` works for now as a loop using `Passage`, subinstances of `Text`, to retrieve the valid informations. Maybe something is more powerfull ?

class `MyCapytain.resources.texts.api.Text` (*urn, resource, citation=None, **kwargs*)
Bases: `MyCapytain.resources.proto.text.Text`

Passage representing object prototype

Parameters

- **urn** (`MyCapytain.common.reference.URN`) – A URN identifier

- **resource** (`MyCapytain.endpoints.proto.CTS`) – An API endpoint
- **citation** (`MyCapytain.resources.texts.tei.Citation`) – Citation for children level
- **id** (`List`) – Identifier of the subreference without URN informations

DEFAULT_LANG = u'eng'

getLabel ()

Retrieve metadata about the text

Return type *Metadata*

Returns Dictionary with label informations

getPassage (*reference=None*)

Retrieve a passage and store it in the object

Parameters **reference** (*MyCapytain.common.reference.Reference* or *List of basestring*) – Reference of the passage

Return type *Passage*

Returns Object representing the passage

Raises *TypeError* when reference is not a list or a *Reference*

getPassagePlus (*reference=None*)

Retrieve a passage and informations around it and store it in the object

Parameters **reference** (*MyCapytain.common.reference.Reference* or *List of basestring*) – Reference of the passage

Return type *Passage*

Returns Object representing the passage

Raises *TypeError* when reference is not a list or a *Reference*

getValidReff (*level=1, reference=None*)

Given a resource, Text will compute valid reffs

Parameters

- **level** (*Int*) – Depth required. If not set, should retrieve first encountered level (1 based)
- **reference** (*Reference*) – Subreference (optional)

Return type *List.basestring*

Returns List of levels

reffs

Get all valid reffs for every part of the Text

Return type *MyCapytain.resources.texts.tei.Citation*

class `MyCapytain.resources.texts.api.Passage` (*urn, resource, *args, **kwargs*)

Bases: *MyCapytain.resources.texts.tei.Passage*

next

Following passage

Return type *Passage*

Returns Following passage at same level

prev

Previous passage

Return type *Passage*

Returns Previous passage at same level

static prevnext (*resource*)

Parse a resource to get the prev and next urn

Parameters **resource** (*etree._Element*) – XML Resource

Returns Tuple representing previous and next urn

Return type (str, str)

2.2 MyCapytain.resources.proto.inventory module

`MyCapytain.resources.proto.inventory.Edition` (*resource=None, urn=None, parents=None*)

class `MyCapytain.resources.proto.inventory.Resource` (*resource=None*)

Bases: `object`

Resource represents any resource from the inventory

export (*format=None*)

parse (*resource*)

Parse the object resource

Parameters **resource** (*Any*) – Resource representing the TextInventory

Return type List

setResource (*resource*)

Set the object property resource

Parameters **resource** (*Any*) – Resource representing the TextInventory

Return type Any

Returns Input resource

class `MyCapytain.resources.proto.inventory.Text` (*resource=None, urn=None, parents=None, subtype='Edition'*)

Bases: `MyCapytain.resources.proto.inventory.Resource`

Represents a CTS Text

class `MyCapytain.resources.proto.inventory.TextGroup` (*resource=None, urn=None, parents=None*)

Bases: `MyCapytain.resources.proto.inventory.Resource`

Represents a CTS Textgroup

class `MyCapytain.resources.proto.inventory.TextInventory` (*resource=None, id=None*)

Bases: `MyCapytain.resources.proto.inventory.Resource`

Represents a CTS Inventory file

`MyCapytain.resources.proto.inventory.Translation` (*resource=None, urn=None, parents=None*)

class `MyCapytain.resources.proto.inventory.Work` (*resource=None, urn=None, parents=None*)
 Bases: `MyCapytain.resources.proto.inventory.Resource`
 Represents a CTS Work

getLang (*key=None*)
 Find a translation with given language

Parameters **key** (*basestring*) – Language to find

Return type [Text]

Returns List of available translations

2.3 MyCapytain.resources.proto.text module

class `MyCapytain.resources.proto.text.Passage` (*parent=None, **kwargs*)
 Bases: `MyCapytain.resources.proto.text.Resource`
 Passage representing object prototype

Parameters

- **urn** (`MyCapytain.common.reference.URN`) – A URN identifier
- **resource** (`lxml.etree._Element`) – A resource
- **parent** (`MyCapytain.resources.texts.tei.Passage`) – Parent of the current passage
- **citation** (`MyCapytain.resources.texts.tei.Citation`) – Citation for children level
- **id** (`List`) – Identifier of the subreference without URN informations

children
 Children of the passage

Return type OrderedDict

Returns Dictionary of children, where key are subreferences

first
 First child of current Passage

Return type None or Passage

Returns None if current Passage has no children, first child passage if available

last
 Last child of current Passage

Return type None or Passage

Returns None if current Passage has no children, last child passage if available

next
 Following passage

Return type *Passage*

Returns Following passage at same level

prev
Previous passage

Return type *Passage*

Returns Previous passage at same level

class `MyCapytain.resources.proto.text.PassagePlus` (*passage, prev, next*)
Bases: `tuple`

next
Alias for field number 2

passage
Alias for field number 0

prev
Alias for field number 1

class `MyCapytain.resources.proto.text.Resource` (*urn=None, resource=None*)
Bases: `object`

Initiate a Resource object

Parameters

- **urn** (`MyCapytain.common.reference.URN`) – A URN identifier
- **resource** (*Any*) – A resource

urn
URN Identifier of the object

Return type *MyCapytain.common.reference.URN*

class `MyCapytain.resources.proto.text.Text` (*citation=None, metadata=None, **kwargs*)
Bases: `MyCapytain.resources.proto.text.Resource`

A CTS Text

citation
Get the lowest cRefPattern in the hierarchy

Return type *MyCapytain.common.reference.Citation*

getLabel ()
Retrieve metadata about the text

Return type `dict`

Returns Dictionary with label informations

getPassage (*reference*)
Retrieve a passage and store it in the object

Parameters **reference** (*MyCapytain.common.reference.Reference* or *List of basestring*) – Reference of the passage

Return type *Passage*

Returns Object representing the passage

Raises *TypeError* when reference is not a list or a Reference

getValidReff (*level=1, reference=None*)
Given a resource, Text will compute valid reffs

Parameters

- **level** (*Int*) – Depth required. If not set, should retrieve first encountered level (1 based)
- **passage** (*Reference*) – Subreference (optional)

Return type List.basestring

Returns List of levels

reffs

Get all valid reffs for every part of the Text

Return type *MyCapytain.resources.texts.tei.Citation*

MyCapytain.endpoints package

3.1 MyCapytain.endpoints.ahab module

class `MyCapytain.endpoints.ahab.Ahab` (*endpoint*)

Bases: `MyCapytain.endpoints.proto.Ahab`

Basic integration of the proto.CTS abstraction

3.2 MyCapytain.endpoints.cts5 module

class `MyCapytain.endpoints.cts5.CTS` (*endpoint, inventory=None*)

Bases: `MyCapytain.endpoints.proto.CTS`

Basic integration of the MyCapytain.endpoints.proto.CTS abstraction

call (*parameters*)

Call an endpoint given the parameters

Parameters `parameters` (*dict*) – Dictionary of parameters

Return type `text`

getCapabilities (*inventory=None*)

Retrieve the inventory information of an API

Parameters `inventory` (*text*) – Name of the inventory

Return type `str`

getFirstUrn (*urn, inventory=None*)

Retrieve the first passage urn of a text

Parameters

- `urn` (*text*) – URN identifying the text
- `inventory` (*text*) – Name of the inventory

Return type `str`

getLabel (*urn, inventory=None*)

Retrieve informations about a CTS Urn

Parameters

- `urn` (*text*) – URN identifying the text's passage (Minimum depth : 1)

- **inventory** (*text*) – Name of the inventory

Return type *str*

getPassage (*urn, inventory=None, context=None*)

Retrieve a passage

Parameters

- **urn** (*text*) – URN identifying the text’s passage (Minimum depth : 1)
- **inventory** (*text*) – Name of the inventory
- **context** (*int*) – Number of citation units at the same level of the citation hierarchy as the requested urn, immediately preceding and immediately following the requested urn to include in the reply

Return type *str*

getPassagePlus (*urn, inventory=None, context=None*)

Retrieve a passage and informations about it

Parameters

- **urn** (*text*) – URN identifying the text’s passage (Minimum depth : 1)
- **inventory** (*text*) – Name of the inventory
- **context** (*int*) – Number of citation units at the same level of the citation hierarchy as the requested urn, immediately preceding and immediately following the requested urn to include in the reply

Return type *str*

getPrevNextUrn (*urn, inventory=None*)

Retrieve the previous and next passage urn of one passage

Parameters

- **urn** (*text*) – URN identifying the text’s passage (Minimum depth : 1)
- **inventory** (*text*) – Name of the inventory

Return type *str*

getValidReff (*urn, inventory=None, level=None*)

Retrieve valid urn-references for a text

Parameters

- **urn** (*text*) – URN identifying the text
- **inventory** (*text*) – Name of the inventory
- **level** (*int*) – Depth of references expected

Return type *str*

3.3 MyCapytain.endpoints.proto module

class MyCapytain.endpoints.proto.**API** (*endpoint*)

Bases: *object*

API Prototype object

Parameters

- **self** (*API*) – Object
- **endpoint** (*text*) – URL of the API

Variables **endpoint** – Url of the endpoint

class `MyCapytain.endpoints.proto.Ahab` (*endpoint*)

Bases: `MyCapytain.endpoints.proto.API`

Abstract Capitains Ahab API See : <http://capitains.github.io/pages/ahab.html>

permalink (*urn, format='xml'*)

Perform a permalink request on API

Return type `str`

search (*query, urn, start=1, limit=5, format='json'*)

Perform a search on given namespace

Parameters

- **query** (*text*) – Term to perform search on
- **urn** (*text*) – Partial or complete urn identifying the request
- **start** (*int*) – Starting element to display
- **limit** (*int*) – Limit of result displayed
- **format** (*str*) – Format to request (json or xml)

Return type `str`

class `MyCapytain.endpoints.proto.CTS` (*endpoint*)

Bases: `MyCapytain.endpoints.proto.API`

CTS API Endpoint Prototype

getCapabilities (*inventory*)

Retrieve the inventory information of an API

Parameters **inventory** (*text*) – Name of the inventory

Return type `str`

getFirstUrn (*urn, inventory*)

Retrieve the first passage urn of a text

Parameters

- **urn** (*text*) – URN identifying the text
- **inventory** (*text*) – Name of the inventory

Return type `str`

getLabel (*urn, inventory*)

Retrieve informations about a CTS Urn

Parameters

- **urn** (*text*) – URN identifying the text's passage (Minimum depth : 1)
- **inventory** (*text*) – Name of the inventory

Return type `str`

getPassage (*urn, inventory, context=None*)

Retrieve a passage

Parameters

- **urn** (*text*) – URN identifying the text’s passage (Minimum depth : 1)
- **inventory** (*text*) – Name of the inventory
- **context** (*int*) – Number of citation units at the same level of the citation hierarchy as the requested urn, immediately preceding and immediately following the requested urn to include in the reply

Return type *str*

getPassagePlus (*urn, inventory, context=None*)

Retrieve a passage and informations about it

Parameters

- **urn** (*text*) – URN identifying the text’s passage (Minimum depth : 1)
- **inventory** (*text*) – Name of the inventory
- **context** (*int*) – Number of citation units at the same level of the citation hierarchy as the requested urn, immediately preceding and immediately following the requested urn to include in the reply

Return type *str*

getPrevNextUrn (*urn, inventory*)

Retrieve the previous and next passage urn of one passage

Parameters

- **urn** (*text*) – URN identifying the text’s passage (Minimum depth : 1)
- **inventory** (*text*) – Name of the inventory

Return type *str*

getValidReff (*urn, inventory, level=1*)

Retrieve valid urn-references for a text

Parameters

- **urn** (*text*) – URN identifying the text
- **inventory** (*text*) – Name of the inventory
- **level** (*int*) – Depth of references expected

Return type *str*

Indices and tables

- [Importing Modules](#)
- [genindex](#)
- [modindex](#)
- [search](#)

m

MyCapytain.common.metadata, 3
MyCapytain.common.reference, 6
MyCapytain.common.utils, 11
MyCapytain.endpoints.ahab, 21
MyCapytain.endpoints.cts5, 21
MyCapytain.endpoints.proto, 22
MyCapytain.resources.proto.inventory,
 16
MyCapytain.resources.proto.text, 17

Symbols

- `__add__()` (MyCapytain.common.metadata.Metadata method), 4
 - `__eq__()` (MyCapytain.common.reference.Reference method), 8
 - `__eq__()` (MyCapytain.common.reference.URN method), 10
 - `__getitem__()` (MyCapytain.common.metadata.Metadata method), 3
 - `__getitem__()` (MyCapytain.common.metadata.Metadata method), 5
 - `__getitem__()` (MyCapytain.common.reference.Reference method), 8
 - `__getitem__()` (MyCapytain.common.reference.URN method), 10
 - `__gt__()` (MyCapytain.common.reference.URN method), 9
 - `__iter__()` (MyCapytain.common.metadata.Metadata method), 4
 - `__iter__()` (MyCapytain.common.metadata.Metadata method), 5
 - `__iter__()` (MyCapytain.common.reference.Citation method), 6
 - `__len__()` (MyCapytain.common.metadata.Metadata method), 4
 - `__len__()` (MyCapytain.common.reference.Citation method), 6
 - `__len__()` (MyCapytain.common.reference.URN method), 9
 - `__lt__()` (MyCapytain.common.reference.URN method), 9
 - `__setitem__()` (MyCapytain.common.metadata.Metadata method), 3
 - `__setitem__()` (MyCapytain.common.metadata.Metadata method), 5
 - `__str__()` (MyCapytain.common.reference.Reference method), 8
 - `__str__()` (MyCapytain.common.reference.URN method), 10
- ## A
- Ahab (class in MyCapytain.endpoints.ahab), 21
 - Ahab (class in MyCapytain.endpoints.proto), 23
 - API (class in MyCapytain.endpoints.proto), 22
- ## C
- `call()` (MyCapytain.endpoints.cts5.CTS method), 21
 - `child` (MyCapytain.common.reference.Citation attribute), 6
 - `children` (MyCapytain.resources.proto.text.Passage attribute), 17
 - Citation (class in MyCapytain.common.reference), 6
 - Citation (class in MyCapytain.resources.texts.tei), 13
 - `citation` (MyCapytain.resources.proto.text.Text attribute), 18
 - `citation` (MyCapytain.resources.texts.local.Text attribute), 14
 - CTS (class in MyCapytain.endpoints.cts5), 21
 - CTS (class in MyCapytain.endpoints.proto), 23
- ## D
- `DEFAULT_LANG` (MyCapytain.resources.texts.api.Text attribute), 15
- ## E
- `Edition()` (in module MyCapytain.resources.proto.inventory), 16
 - `export()` (MyCapytain.resources.proto.inventory.Resource method), 16
- ## F
- `fill()` (MyCapytain.common.reference.Citation method), 7
 - `first` (MyCapytain.resources.proto.text.Passage attribute), 17
- ## G
- `getCapabilities()` (MyCapytain.endpoints.cts5.CTS method), 21

getCapabilities() (MyCapytain.endpoints.proto.CTS method), 23

getFirstUrn() (MyCapytain.endpoints.cts5.CTS method), 21

getFirstUrn() (MyCapytain.endpoints.proto.CTS method), 23

getLabel() (MyCapytain.endpoints.cts5.CTS method), 21

getLabel() (MyCapytain.endpoints.proto.CTS method), 23

getLabel() (MyCapytain.resources.proto.text.Text method), 18

getLabel() (MyCapytain.resources.texts.api.Text method), 15

getLang() (MyCapytain.resources.proto.inventory.Work method), 17

getPassage() (MyCapytain.endpoints.cts5.CTS method), 22

getPassage() (MyCapytain.endpoints.proto.CTS method), 23

getPassage() (MyCapytain.resources.proto.text.Text method), 18

getPassage() (MyCapytain.resources.texts.api.Text method), 15

getPassage() (MyCapytain.resources.texts.local.Text method), 14

getPassagePlus() (MyCapytain.endpoints.cts5.CTS method), 22

getPassagePlus() (MyCapytain.endpoints.proto.CTS method), 24

getPassagePlus() (MyCapytain.resources.texts.api.Text method), 15

getPassagePlus() (MyCapytain.resources.texts.local.Text method), 14

getPrevNextUrn() (MyCapytain.endpoints.cts5.CTS method), 22

getPrevNextUrn() (MyCapytain.endpoints.proto.CTS method), 24

getValidReff() (MyCapytain.endpoints.cts5.CTS method), 22

getValidReff() (MyCapytain.endpoints.proto.CTS method), 24

getValidReff() (MyCapytain.resources.proto.text.Text method), 18

getValidReff() (MyCapytain.resources.texts.api.Text method), 15

getValidReff() (MyCapytain.resources.texts.local.Text method), 14

I

ingest() (MyCapytain.resources.texts.tei.Citation static method), 13

L

last (MyCapytain.resources.proto.text.Passage attribute), 17

M

Metadata (class in MyCapytain.common.metadata), 3

Metadatum (class in MyCapytain.common.metadata), 4

MyCapytain.common.metadata (module), 3

MyCapytain.common.reference (module), 6, 11

MyCapytain.common.utils (module), 11

MyCapytain.endpoints.ahab (module), 21

MyCapytain.endpoints.cts5 (module), 21

MyCapytain.endpoints.proto (module), 22

MyCapytain.resources.proto.inventory (module), 16

MyCapytain.resources.proto.text (module), 17

N

name (MyCapytain.common.reference.Citation attribute), 7

next (MyCapytain.resources.proto.text.Passage attribute), 17

next (MyCapytain.resources.proto.text.PassagePlus attribute), 18

next (MyCapytain.resources.texts.api.Passage attribute), 15

normalize() (in module MyCapytain.common.utils), 11

NS (in module MyCapytain.common.utils), 11

P

parent (MyCapytain.common.reference.Reference attribute), 9

parse() (MyCapytain.resources.proto.inventory.Resource method), 16

Passage (class in MyCapytain.resources.proto.text), 17

Passage (class in MyCapytain.resources.texts.api), 15

Passage (class in MyCapytain.resources.texts.tei), 13

passage (MyCapytain.resources.proto.text.PassagePlus attribute), 18

PassagePlus (class in MyCapytain.resources.proto.text), 18

permalink() (MyCapytain.endpoints.proto.Ahab method), 23

prev (MyCapytain.resources.proto.text.Passage attribute), 17

prev (MyCapytain.resources.proto.text.PassagePlus attribute), 18

prev (MyCapytain.resources.texts.api.Passage attribute), 15

prevnext() (MyCapytain.resources.texts.api.Passage static method), 16

R

REF_REPLACER() (in module MyCapytain.common.reference), 7

Reference (class in MyCapytain.common.reference), 7
 reffs (MyCapytain.resources.proto.text.Text attribute), 19
 reffs (MyCapytain.resources.texts.api.Text attribute), 15
 refsDecl (MyCapytain.common.reference.Citation attribute), 7
 Resource (class in MyCapytain.resources.proto.inventory), 16
 Resource (class in MyCapytain.resources.proto.text), 18

S

scope (MyCapytain.common.reference.Citation attribute), 7
 search() (MyCapytain.endpoints.proto.Ahab method), 23
 setDefault() (MyCapytain.common.metadata.Metadata method), 6
 setResource() (MyCapytain.resources.proto.inventory.Resource method), 16

T

Text (class in MyCapytain.resources.proto.inventory), 16
 Text (class in MyCapytain.resources.proto.text), 18
 Text (class in MyCapytain.resources.texts.api), 14
 Text (class in MyCapytain.resources.texts.local), 13
 text() (MyCapytain.resources.texts.tei.Passage method), 13
 TextGroup (class in MyCapytain.resources.proto.inventory), 16
 TextInventory (class in MyCapytain.resources.proto.inventory), 16
 Translation() (in module MyCapytain.resources.proto.inventory), 16

U

URN (class in MyCapytain.common.reference), 9
 urn (MyCapytain.resources.proto.text.Resource attribute), 18

W

Work (class in MyCapytain.resources.proto.inventory), 16

X

xml (MyCapytain.resources.texts.tei.Passage attribute), 13
 xmlparser() (in module MyCapytain.common.utils), 11
 xpath (MyCapytain.common.reference.Citation attribute), 7