
pyIUDX

Release 1

Apr 20, 2020

Contents

1	pyIUDX.rs package	1
1.1	Submodules	1
1.2	pyIUDX.rs.item module	1
1.3	pyIUDX.rs.rs module	4
1.4	Module contents	8
2	pyIUDX.cat package	9
2.1	Submodules	9
2.2	pyIUDX.cat.cat module	9
2.3	Module contents	10
3	pyIUDX.auth package	11
3.1	Submodules	11
3.2	pyIUDX.auth.auth module	11
3.3	Module contents	12
4	Indices and tables	13
	Python Module Index	15
	Index	17

1.1 Submodules

1.2 pyIUDX.rs.item module

class pyIUDX.rs.item.GeoProperty (*name*)

Bases: object

Container class for a Geo property GeoProperty is a spatial property having further attributes such as coordinates for either a point or a polygon.

name = None

Type in case geoProperty is static

reset ()

Reset coordinates This is only invoked in case where this geoProperty is time varying TODO: Workaround for type

setDynamicGeo (*time, coordinates*)

Set Value for this geoProperty

To be used only when this geoProperty is time varying

Parameters

- **time** (*datetime.datetime ()*) – Time index
- **value** (*float*) – Value

setStaticGeo (*geoProperty*)

Static points

sort ()

Sort time series

class pyIUDX.rs.item.Item (*catUrl, rsUrl, resourceItemId, dataModel=None*)

Bases: object

class for an iudx resource item

A resource item has it's static attribute representation in a catalogue and a dynamic "data" representation in a resource server. This class presents an abstraction layer combining both

cat = None

get rs from catalogue item

Type TODO

during (*start*, *end*)

Get data during a set interval for all properties belonging to this item

Parameters

- **startTime** (*string*) – Start time
- **endTime** (*string*) – End time

Returns numpy 2d array with 0th column as time

Return type value (ndarray)

latest ()

Get latest data for all properties belonging to this item

Returns Returns back the updated object

Return type self (object)

latestWith (*attr*, *val*)

Get latest data for all properties belonging to this item with a specific attribute

This method will give latest data where an attribute is specified. For e.g attr = "ROUTE_ID", val = "110" will give latest data for that bus

Parameters

- **attr** (*string*) – The name of the attribute
- **val** (*string*) – The value of the attribute

Returns Returns back the updated object

Return type self (object)

num_time_indices = None

Read item properties

populateValue (*data*)

Helper function to populate a QuantitativeProperty's value array

quantitativeProperties = None

Load datamodel properties

reset ()

Reset data of all quantitative attributes of this item :returns: Returns back the updated object :rtype: self (object)

valueBetween (*attrName*, *minval*, *maxVal*)

Get all data during which this attribute was between min and max val

Parameters

- **attrName** (*string*) – name of the attribute

- **minVal** (*int*) – minimum value
- **minVal** – maximum value

class pyIUDX.rs.item.**Items** (*catUrl, rsUrl, items=None*)

Bases: collections.abc.MutableSequence

class for a list of iudx resource items.

This class extends a list to provide Class Item style functionality coupled with multiprocessing pool to allow for faster data access

during (*startTime, endTime*)

Get data during a set interval

For all resource items of this instance and for all properties belonging to this item

Parameters

- **startTime** (*string*) – Start time
- **endTime** (*string*) – End time

Returns numpy 2d array with 0th column as time

Return type value (ndarray)

getDuring (*obj, startTime, endTime*)

Multiprocessed job

getLatest (*obj*)

Multiprocessed job

initItem (*catUrl, rsUrl, item, objList*)

Multiprocessed job

insert (*ii, val*)

S.insert(index, value) – insert value before index

latest ()

Get latest data

For all resource items of this instance and for all properties belonging to this item

Returns Returns back the updated object

Return type self (object)

class pyIUDX.rs.item.**Property** (*name, properties*)

Bases: object

Container class for a Property

Property is an aspect of a resource item that describes it or its current value

reset ()

setValue (*value, time=None*)

Set State for this Property :param value: Value :type value: string :param time: Time index :type time: datetime.datetime()

sort ()

Sort time series

class pyIUDX.rs.item.**QuantitativeProperty** (*obj, name, properties*)

Bases: object

Container class for a quantitative property `QuantitativeProperty` is a measurable property having further attributes such as units. Values are always indexed with time.

during (*startTime, endTime*)

Get data during a set interval for this property :param startTime: Start time :type startTime: string :param endTime: End time :type endTime: string

Returns numpy 2d array with 0th column as time

Return type value (ndarray)

latest ()

Get latest data for this property :returns: numpy 2d array with 0th column as time :rtype: value (ndarray)

reset (*num_time_indices*)

Reset value of this Property

setValue (*time, value*)

Set Value for this Property :param time: Time index :type time: datetime.datetime() :param value: Value :type value: float

sort (*updatedAtIdx*)

Sort time series

valueBetween (*minval, maxVal*)

Get all data during which this attribute was between min and max val :param minVal: minimum value :type minVal: int :param minVal: maximum value :type minVal: int

Returns numpy 2d array with 0th column as time

Return type value (ndarray)

1.3 pyIUDX.rs.rs module

class `pyIUDX.rs.rs.ResourceServer` (*rsUrl, cert=None, key=None*)

Bases: object

dispParams ()

Display rs initialization parameter

Returns version string

Return type (rsDomain, rsPort, rsVersion) (string, string, string)

download (*url, data*)

Use requests library to make a search request

Returns Response body

Return type resp (object)

downloadData (*groupId, opts=None, token=None*)

Download data from a resource server

An optional options dictionary can be passed to get more specific data. The options dictionary follows the schema: <https://raw.githubusercontent.com/iudx/pyIUDX/rs/pyIUDX/rs/opts.json>

Parameters

- **groupId** (*string*) – id of the resource item
- **opts** (*Dict*) – dictionary of various options

Returns rs constructed url

Return type url (string)

getData (*id*, *opts=None*, *token=None*)

Get data from a resource server

An optional options dictionary can be passed to get more specific data. The options dictionary follows the schema: <https://raw.githubusercontent.com/iudx/pyIUDX/rs/pyIUDX/rs/opts.json>

Parameters

- **id** (*string*) – id of the resource item
- **opts** (*Dict*) – dictionary of various options

Returns rs constructed url

Return type url (string)

getDataAfter (*id*, *time*, *token=None*)

Get data after a given time

Parameters

- **id** (*string*) – id of the resource item
- **time** (*string*) – Starting from

Returns

Array with a time indexed dictionary item corresponding to the data

Return type data (List[Dict])

getDataAroundDuring (*id*, *point*, *radius*, *startTime*, *endTime*, *token=None*)

Get data around a specific point(lat, lon) and radius(meters) and during a time

Parameters

- **id** (*string*) – id of the resource item
- **point** (*List[Float]*) – point [lat, lon]
- **radius** (*int*) – radius in meters
- **startTime** (*string*) – Starting from
- **endTime** (*string*) – Till

Returns

Array with a time indexed dictionary item corresponding to the data

Return type data (List[Dict])

getDataBefore (*id*, *time*, *token=None*)

Get data before a given time

Parameters

- **id** (*string*) – id of the resource item
- **time** (*string*) – Ending at

Returns

Array with a time indexed dictionary item corresponding to the data

Return type data (List[Dict])

getDataDuring (*id*, *startTime*, *endTime*, *token=None*)

Get data during a time interval

Parameters

- **id** (*string*) – id of the resource item
- **startTime** (*string*) – Starting from
- **endTime** (*string*) – Till

Returns

Array with a time indexed dictionary item corresponding to the data

Return type data (List[Dict])

getDataValuesBetween (*id*, *attribute*, *minVal*, *maxVal*, *token=None*)

Get data of an item for which an attribute is between minVal and maxVal

Parameters

- **id** (*string*) – id of the resource item
- **attribute** (*string*) – attribute name
- **minVal** (*float*) – minimum value
- **maxVal** (*float*) – maximum value

Returns

Array with a time indexed dictionary item corresponding to the data

Return type data (List[Dict])

getDataValuesGreater (*id*, *attribute*, *minVal*, *token=None*)

Get data of an item for which an attribute is greater than minVal

Parameters

- **id** (*string*) – id of the resource item
- **attribute** (*string*) – attribute name
- **minVal** (*float*) – minimum value

Returns

Array with a time indexed dictionary item corresponding to the data

Return type data (List[Dict])

getDataValuesLesser (*id*, *attribute*, *maxVal*, *token=None*)

Get data of an item for which an attribute is lesser than maxVal

Parameters

- **id** (*string*) – id of the resource item
- **attribute** (*string*) – attribute name
- **maxVal** (*float*) – maximum value

Returns

Array with a time indexed dictionary item corresponding to the data

Return type data (List[Dict])

getDataValuesLikeDuring (*id*, *attribute*, *val*, *startTime*, *endTime*, *token=None*)

Get data of an item for which an attribute is like value between a time

Parameters

- **id** (*string*) – id of the resource item
- **attribute** (*string*) – attribute name
- **val** (*string*) – value

Returns

Array with a time indexed dictionary item corresponding to the data

Return type data (List[Dict])

getLatestData (*id*, *token=None*)

Get latest data

Parameters **id** (*string*) – id of the resource item

Returns

Array with a single dictionary item corresponding to the data

Return type data (List[Dict])

getLatestDataAround (*id*, *point*, *radius*, *token=None*)

Get data around a specific point(lat, lon) and radius(meters)

Parameters

- **id** (*string*) – id of the resource item
- **point** (*List[Float]*) – point [lat, lon]
- **radius** (*int*) – radius in meters

Returns

Array with a time indexed dictionary item corresponding to the data

Return type data (List[Dict])

getLatestDataAroundLike (*id*, *point*, *radius*, *attributeName*, *attributeValue*, *token=None*)

Get data around a specific point(lat, lon) and radius(meters) which has an attribute like

Parameters

- **id** (*string*) – id of the resource item
- **point** (*List[Float]*) – point [lat, lon]
- **radius** (*int*) – radius in meters

Returns

Array with a time indexed dictionary item corresponding to the data

Return type data (List[Dict])

getLatestDataValuesLike (*id*, *attribute*, *val*, *token=None*)

Get latest data of an item for which an attribute is like value

Parameters

- **id** (*string*) – id of the resource item
- **attribute** (*string*) – attribute name

- **val** (*string*) – value

Returns

Array with a time indexed dictionary item corresponding to the data

Return type data (List[Dict])

getStatus (*id*, *token=None*)

Get Status of a resource item

Parameters **id** (*string*) – id of the resource item

Returns True if up

Return type status (bool)

TODO:

getUrl ()

Get rs constructed url

Returns rs constructed url

Return type url (string)

search (*url*, *data*)

Use requests library to make a search request

Returns Response body

Return type resp (object)

1.4 Module contents

2.1 Submodules

2.2 pyIUDX.cat.cat module

class `pyIUDX.cat.cat.Catalogue` (*catUrl*)

Bases: `object`

checkConnection ()

getAllItems ()

Get all catalogue items :returns: List of catalogue items (dicts) :rtype: list (List[Dict])

getDataModel (*id*)

Get the data model for a given id :returns: List of catalogue items (dicts) :rtype: list (List[Dict])

getItemCount (*attributes=None, filters=None, geo=None*)

Number of items matching the criterion :param attributes: Array of key value pairs

For e.x, {“tags”: [“a”, “b”], “provider”: [“c”]}

Parameters **filters** (*List[str]*) – Array of strings as filter opts For e.x, [“id”, “provider”]

Returns number of items or -1 if fail

Return type count (int)

getManyResourceItems (*attributes=None, filters=None, geo=None*)

Items matching the criterion :param attributes: Array of key value pairs

For e.x, {“tags”: [“a”, “b”], “provider”: [“c”]}

Parameters **filters** (*List[str]*) – Array of strings as filter opts For e.x, [“id”, “provider”]

Returns List of catalogue items (dicts)

Return type list (List[Dict])

getOneResourceItem (*id*, *filters=None*)

Item given the id :param id: ID of the resourceItem :type id: string :param filters: Array of strings as filter opts

For e.x, ["id", "provider"]

Returns A catalogue items (dicts)

Return type item (Dict)

makeOpts (*attributes=None*, *filters=None*, *geo=None*)

Make attributes options string :param attributes: Array of key value pairs

For e.x, {"tags": ["a", "b"], "provider": ["c"]}

Parameters **filters** (*List[str]*) – Array of strings as filter opts For e.x, ["id", "provider"]

Returns options as a string for a GET method

Return type opts (string)

2.3 Module contents

3.1 Submodules

3.2 pyIUDX.auth.auth module

```
class pyIUDX.auth.auth.Auth (certificate, key, auth_server='auth.iudx.org.in', version=1)
    Bases: object

    add_consumer_to_group (consumer, group, valid_till)
    append_policy (policy)
    audit_tokens (hours)
    call (api, body=None)
    delete_consumer_from_group (consumer, group)
    get_certificate_info ()
    get_policy ()
    get_token (request, token_time=None, existing_token=None)
    introspect_token (token, server_token=None)
    list_group (consumer, group=None)
    revert_policy ()
    revoke_all (serial, fingerprint)
    revoke_token_hashes (token_hashes)
    revoke_tokens (tokens)
    set_policy (policy)
```

3.3 Module contents

CHAPTER 4

Indices and tables

- `genindex`
- `modindex`
- `search`

p

- `pyIUDX.auth`, 12
- `pyIUDX.auth.auth`, 11
- `pyIUDX.cat`, 10
- `pyIUDX.cat.cat`, 9
- `pyIUDX.rs`, 8
- `pyIUDX.rs.item`, 1
- `pyIUDX.rs.rs`, 4

A

add_consumer_to_group() (pyIUDX.auth.auth.Auth method), 11
 append_policy() (pyIUDX.auth.auth.Auth method), 11
 audit_tokens() (pyIUDX.auth.auth.Auth method), 11
 Auth (class in pyIUDX.auth.auth), 11

C

call() (pyIUDX.auth.auth.Auth method), 11
 cat (pyIUDX.rs.item.Item attribute), 2
 Catalogue (class in pyIUDX.cat.cat), 9
 checkConnection() (pyIUDX.cat.cat.Catalogue method), 9

D

delete_consumer_from_group() (pyIUDX.auth.auth.Auth method), 11
 dispParams() (pyIUDX.rs.rs.ResourceServer method), 4
 download() (pyIUDX.rs.rs.ResourceServer method), 4
 downloadData() (pyIUDX.rs.rs.ResourceServer method), 4
 during() (pyIUDX.rs.item.Item method), 2
 during() (pyIUDX.rs.item.Items method), 3
 during() (pyIUDX.rs.item.QuantitativeProperty method), 4

G

GeoProperty (class in pyIUDX.rs.item), 1
 get_certificate_info() (pyIUDX.auth.auth.Auth method), 11
 get_policy() (pyIUDX.auth.auth.Auth method), 11
 get_token() (pyIUDX.auth.auth.Auth method), 11
 getAllItems() (pyIUDX.cat.cat.Catalogue method), 9
 getData() (pyIUDX.rs.rs.ResourceServer method), 5

getDataAfter() (pyIUDX.rs.rs.ResourceServer method), 5
 getDataAroundDuring() (pyIUDX.rs.rs.ResourceServer method), 5
 getDataBefore() (pyIUDX.rs.rs.ResourceServer method), 5
 getDataDuring() (pyIUDX.rs.rs.ResourceServer method), 5
 getDataModel() (pyIUDX.cat.cat.Catalogue method), 9
 getDataValuesBetween() (pyIUDX.rs.rs.ResourceServer method), 6
 getDataValuesGreater() (pyIUDX.rs.rs.ResourceServer method), 6
 getDataValuesLesser() (pyIUDX.rs.rs.ResourceServer method), 6
 getDataValuesLikeDuring() (pyIUDX.rs.rs.ResourceServer method), 6
 getDuring() (pyIUDX.rs.item.Items method), 3
 getItemCount() (pyIUDX.cat.cat.Catalogue method), 9
 getLatest() (pyIUDX.rs.item.Items method), 3
 getLatestData() (pyIUDX.rs.rs.ResourceServer method), 7
 getLatestDataAround() (pyIUDX.rs.rs.ResourceServer method), 7
 getLatestDataAroundLike() (pyIUDX.rs.rs.ResourceServer method), 7
 getLatestDataValuesLike() (pyIUDX.rs.rs.ResourceServer method), 7
 getManyResourceItems() (pyIUDX.cat.cat.Catalogue method), 9
 getOneResourceItem() (pyIUDX.cat.cat.Catalogue method), 10
 getStatus() (pyIUDX.rs.rs.ResourceServer method), 8
 getUrl() (pyIUDX.rs.rs.ResourceServer method), 8

I

initItem() (pyIUDX.rs.item.Items method), 3

`insert()` (*pyIUDX.rs.item.Items method*), 3
`introspect_token()` (*pyIUDX.auth.auth.Auth method*), 11
`Item` (*class in pyIUDX.rs.item*), 1
`Items` (*class in pyIUDX.rs.item*), 3

L

`latest()` (*pyIUDX.rs.item.Item method*), 2
`latest()` (*pyIUDX.rs.item.Items method*), 3
`latest()` (*pyIUDX.rs.item.QuantitativeProperty method*), 4
`latestWith()` (*pyIUDX.rs.item.Item method*), 2
`list_group()` (*pyIUDX.auth.auth.Auth method*), 11

M

`makeOpts()` (*pyIUDX.cat.cat.Catalogue method*), 10

N

`name` (*pyIUDX.rs.item.GeoProperty attribute*), 1
`num_time_indices` (*pyIUDX.rs.item.Item attribute*), 2

P

`populateValue()` (*pyIUDX.rs.item.Item method*), 2
`Property` (*class in pyIUDX.rs.item*), 3
`pyIUDX.auth` (*module*), 12
`pyIUDX.auth.auth` (*module*), 11
`pyIUDX.cat` (*module*), 10
`pyIUDX.cat.cat` (*module*), 9
`pyIUDX.rs` (*module*), 8
`pyIUDX.rs.item` (*module*), 1
`pyIUDX.rs.rs` (*module*), 4

Q

`quantitativeProperties` (*pyIUDX.rs.item.Item attribute*), 2
`QuantitativeProperty` (*class in pyIUDX.rs.item*), 3

R

`reset()` (*pyIUDX.rs.item.GeoProperty method*), 1
`reset()` (*pyIUDX.rs.item.Item method*), 2
`reset()` (*pyIUDX.rs.item.Property method*), 3
`reset()` (*pyIUDX.rs.item.QuantitativeProperty method*), 4
`ResourceServer` (*class in pyIUDX.rs.rs*), 4
`revert_policy()` (*pyIUDX.auth.auth.Auth method*), 11
`revoke_all()` (*pyIUDX.auth.auth.Auth method*), 11
`revoke_token_hashes()` (*pyIUDX.auth.auth.Auth method*), 11
`revoke_tokens()` (*pyIUDX.auth.auth.Auth method*), 11

S

`search()` (*pyIUDX.rs.rs.ResourceServer method*), 8
`set_policy()` (*pyIUDX.auth.auth.Auth method*), 11
`setDynamicGeo()` (*pyIUDX.rs.item.GeoProperty method*), 1
`setStaticGeo()` (*pyIUDX.rs.item.GeoProperty method*), 1
`setValue()` (*pyIUDX.rs.item.Property method*), 3
`setValue()` (*pyIUDX.rs.item.QuantitativeProperty method*), 4
`sort()` (*pyIUDX.rs.item.GeoProperty method*), 1
`sort()` (*pyIUDX.rs.item.Property method*), 3
`sort()` (*pyIUDX.rs.item.QuantitativeProperty method*), 4

V

`valueBetween()` (*pyIUDX.rs.item.Item method*), 2
`valueBetween()` (*pyIUDX.rs.item.QuantitativeProperty method*), 4