
SEEK Documentation

Release 1.0

Bogdan Gherasim

Jun 12, 2020

Contents:

1	Indices and tables	5
	Python Module Index	7
	Index	9

Package SEEK - THE SEEK NOTEBOOK

This package provides methods and functions implementing the SEEK API to be used in bioinformatic context.

Build for intuitive usage, this package allows the user to retrieve data from the web-based cataloguing platform The Fairdom Hub powered by SEEK in order to be used in the Jupyter Notebook web app, supporting widgets for user interfacing. _____

Third Year Project, Bogdan Gherasim, The University of Manchester, 2019

`SEEK.auth()`

Method used externally to get FairdomHub credentials from user

Returns (username, password)

Return type tuple

class `SEEK.read(auth=None)`

This class provides methods and functions for reading and browsing the SEEK

Parameters `auth(SEEK.auth())` – FairdomHub credentials; can be left empty

APISearch()

Used for Demo Presentation. Searches without sophisticated interpretation. Process the request and retrieve the JSON

Example

```
>>> import SEEK as S
>>> search = S.read(auth)
>>> search.searchAdvancedSetup()
Search multithreading
Decide how fast will the search run
Search results are usually less and the search requires less to be
requested at once.
Relationships are usually a lot more, therefore it requires more
requests at once
How many search results should be requested per thread:
>>> 1
How many relationships should be requested per thread:
>>> 1
```

browse()

Runs a browse in the API

This method prints the results before returning

Returns whether the search is successful or not

Return type bool

createRelationshipList()

Create the relationship list by parsing the search result list

This method fills the relationshipList array attribute before returning

Returns number of relationships found

Return type int

createRequestList()

Creates the list of requests by parsing a RAW Search Result JSON by searching for each pair of (id, type) present in the string.

Returns list of tuples

Return type list

Example

```
>>> import SEEK as S
>>> search = S.read(auth)
>>> search.APISearch()
...
>>> search.createRequestList()
>>> print(str(len(s.requestList)))
132
```

download()

Downloads a content blob in the file

getRelationshipsFrom(request)

Creates the relationships list by parsing the data attribute of the specified object

Parameters **request** (*SEEK.read*) – object to extract relationships from

Returns unprocessed relationships

Return type list of tuples

parallelRequest(requests, requestPerThread)

Creates a list of processed request from an unprocessed list

Uses multithreading to read a number of request and retrieve results from the API

Parameters

- **requests** (*list*) – list of request to process
- **requestPerThread** (*int*) – specifies how many requests each thread is supposed to handle and, therefore, how many threads are there going to be created

printBrowse()

Prints the results of a browse

printSearch()

Prints a search result

removeDuplicateRelationships()

Loops through the relationshipList array attribute and removes the duplicated entries for further modifications

search(TYPE, ID)

Searches and retrieves a specific object from the API along with it's relationships

Parameters

- **TYPE** (*string*) – eg: assays / studies / investigations etc.
- **ID** (*int*) – identification number

Returns whether the search is successful or not

Return type bool

searchAdvancedSetup()

Used to set up how fast the multithreading runs

The method prompts how many requests there will be in each thread, eg: if there are 100 requests and you run this with 1 requests/thread, there will be loads of multithreading(100 threads) and therefore the process will run really fast... or at least will try to. If you run it with 50 requests/thread it will run slow(2 threads).

Example

```
>>> import SEEK as S
>>> search = S.read(auth)
>>> search.searchAdvancedSetup()
Search multithreading
Decide how fast will the search run
Search results are usually less and the search requires less to be
requested at once.
Relationships are usually a lot more, therefore it requires more
requests at once
How many search results should be requested per thread:
>>> 1
How many relationships should be requested per thread:
>>> 1
```

substituteRelationships (*relationshipsList*)

Loops through the search results and substitutes the raw relationships with requested objects

This method replaces the relationships from the `data.relationships` attribute with objects from the first parameter

Parameters **relationshipsList** (*list*) – the processed relationships used to substitute the raw ones

substituteRelationshipsForSearchResults (*relationshipsList*)

Used when browsing

view (*columnForHeader, page*)

View the downloaded file

class **SEEK.write** (*auth=None, DNS='https://testing.sysmo-db.org/'*)

This class provides methods and functions for writing into the SEEK

Parameters

- **auth** (*SEEK.auth()*) – FairdomHub credentials; can be left empty
- **DNS** (*string*) – Specify if you are posting to the official HUB or leave empty to write to testing.sysmo-db

fillDescription ()

Mandatory

fillSEEKForm ()

Mandatory

post ()

Uses python requests to post the object created previously and saved in the JSON attribute to the SEEK

Returns whether the post is successful or not

Return type bool

Example

```
>>> import SEEK as S
>>> request = S.write()
>>> request.post()
False
>>> request.selectResearchType()
...
```

(continues on next page)

(continued from previous page)

```
>>> request.selectAssayKind()  
...  
>>> request.fillDescription()  
...  
>>> request.selectPolicyAccess()  
...  
>>> request.fillSEEKForm()  
...  
>>> request.post()  
True
```

selectAssayKind()
Mandatory for assays

selectPolicyAccess()
Mandatory

selectResearchType()
Mandatory

CHAPTER 1

Indices and tables

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

S

SEEK, ??

A

APISearch() (*SEEK.read method*), 1
auth() (*in module SEEK*), 1

B

browse() (*SEEK.read method*), 1

C

createRelationshipList() (*SEEK.read method*), 1
createRequestList() (*SEEK.read method*), 1

D

download() (*SEEK.read method*), 2

F

fillDescription() (*SEEK.write method*), 3
fillSEEKForm() (*SEEK.write method*), 3

G

getRelationshipsFrom() (*SEEK.read method*), 2

P

parallelRequest() (*SEEK.read method*), 2
post() (*SEEK.write method*), 3
printBrowse() (*SEEK.read method*), 2
printSearch() (*SEEK.read method*), 2

R

read (*class in SEEK*), 1
removeDuplicateRelationships() (*SEEK.read method*), 2

S

search() (*SEEK.read method*), 2
searchAdvancedSetup() (*SEEK.read method*), 2
SEEK (*module*), 1
selectAssayKind() (*SEEK.write method*), 4

selectPolicyAccess() (*SEEK.write method*), 4
selectResearchType() (*SEEK.write method*), 4
substituteRelationships() (*SEEK.read method*), 3
substituteRelationshipsForSearchResults() (*SEEK.read method*), 3

V

view() (*SEEK.read method*), 3

W

write (*class in SEEK*), 3