BaseSpacePy Documentation

Release 0.1.2

Morten Kallberg

October 02, 2012

CONTENTS

AVAILABLE MODULES

1.1 API

class `BaseSpacePy.api.BaseSpaceAPI.`**`BaseSpaceAPI`**(*clientKey*, *clientSecret*, *apiServer*, *version*, *appSessionId*, *AccessToken=''*)

   The main API class used for all communication with with the REST server

   `appResultFileUpload`(*Id*, *localPath*, *fileName*, *directory*, *contentType*, *multipart=0*)

   Uploads a file associated with an AppResult to BaseSpace and returns the corresponding file object

   Parameters

   - **Id** – AppResult id.

   - **localPath** – The local path to the file to be uploaded.

   - **fileName** – The desired filename in the AppResult folder on the BaseSpace server.

   - **directory** – The directory the file should be placed in.

   - **contentType** – The content-type of the file.

   `createAppResult`(*Id*, *name*, *desc*, *samples=*[ ], *appSessionId=None*)

   Create an AppResult object

   Parameters

   - **Id** – The id of the project in which the AppResult is to be added

   - **name** – The name of the AppResult

   - **desc** – A describtion of the AppResult

   - **samples** – (Optional) The samples

   - **appSessionId** – (Optional) If no appSessionId is given, the id used to initialize the BaseSpaceAPI instance

   will be used. If appSessionId is set equal to an empty string, a new appsession will be created for the

   `fileDownload`(*Id*, *localDir*, *name*, *range=*[ ])

   Downloads a BaseSpace file to a local directory

   Parameters

   - **Id** – The file id

   - **localDir** – The local directory to place the file in

   - **name** – The name of the local file

- **range** – (Optional) The byte range of the file to retrieve (not yet implemented)

filterVariantSet (*Id*, *Chrom*, *StartPos*, *EndPos*, *Format*, *queryPars={'Limit': '100', 'SortBy': 'Position', 'SortDir': 'Asc', 'Offset': '0'}*)

    List the variants in a set of variants. Maximum returned records is 1000

        Parameters

- **Id** – The id of the variant file
- **Chrom** – The chromosome of interest
- **StartPos** – The start position of the sequence of interest
- **EndPos** – The start position of the sequence of interest
- **Format** – Set to 'vcf' to get the results as lines in VCF format
- **queryPars** – An (optional) object of type QueryParameters for custom sorting and filtering

getAccess (*obj*, *accessType='write'*, *web=0*, *redirectURL=''*, *state=''*)

        Parameters

- **obj** – The data object we wish to get access to
- **accessType** – (Optional) the type of access (read|write), default is write
- **web** – (Optional) true if the App is web-based, default is false meaning a device based app
- **redirectURL** – (Optional) For the web-based case, a
- **state** – (Optional)

getAccessToken ()

    Returns the access-token that was used to initialize the BaseSpaceAPI object.

getAccessibleRunsByUser (*Id*, *queryPars={'Limit': '100', 'SortBy': 'Id', 'SortDir': 'Asc', 'Offset': '0'}*)

    Returns a list of accessible runs for the User with id=Id

        Parameters

- **Id** – An user id
- **queryPars** – An (optional) object of type QueryParameters for custom sorting and filtering

getAppResultById (*Id*)

    Returns an AppResult object corresponding to Id

        Parameters Id – The Id of the AppResult

getAppResultFiles (*Id*, *queryPars={'Limit': '100', 'SortBy': 'Id', 'SortDir': 'Asc', 'Offset': '0'}*)

    Returns a list of File object for the AppResult with id = Id

        Parameters

- **Id** – The id of the appresult.
- **queryPars** – An (optional) object of type QueryParameters for custom sorting and filtering

getAppResultsByProject (*Id*, *queryPars={'Limit': '100', 'SortBy': 'Id', 'SortDir': 'Asc', 'Offset': '0'}*, *statuses=[ ]*)

    Returns a list of AppResult object associated with the project with Id

        Parameters

- **Id** – The project id

> • **queryPars** – An (optional) object of type QueryParameters for custom sorting and filtering
>
> • **statuses** – An (optional) list of AppResult statuses to filter by

getAppSession(*Id=''*)

Returns an AppSession instance containing user and data-type the app was triggered by/on :param Id: (Optional) The AppSessionId, id not supplied the AppSessionId used for instantiating the BaseSpaceAPI instance.

> **Parameters Id** – (Optional) AppSession id, if not supplied the AppSession id used to initialize the

getAppSessionById(*id*)

Returns the appSession identified by id

> **Parameters id** – The id of the appSession

getAvailableGenomes(*queryPars={'Limit': '100', 'SortBy': 'Id', 'SortDir': 'Asc', 'Offset': '0'}*)

Returns a list of all available genomes

> **Parameters queryPars** – An (optional) object of type QueryParameters for custom sorting and filtering

getCoverageMetaInfo(*Id*, *Chrom*)

Returns Metadata about coverage as a CoverageMetadata instance

> **Parameters**
>
> > • **Id** – he Id of the Bam file
> >
> > • **Chrom** – Chromosome to query

getFileById(*Id*)

Returns a file object by Id

> **Parameters Id** – The id of the file

getFilesBySample(*Id*, *queryPars={'Limit': '100', 'SortBy': 'Id', 'SortDir': 'Asc', 'Offset': '0'}*)

Returns a list of File objects associated with sample with Id

> **Parameters**
>
> > • **Id** – A Sample id
> >
> > • **queryPars** – An (optional) object of type QueryParameters for custom sorting and filtering

getGenomeById(*Id*)

Returns an instance of Genome with the specified Id

> **Parameters Id** – The genome id

getIntervalCoverage(*Id*, *Chrom*, *StartPos=None*, *EndPos=None*)

Mean coverage levels over a sequence interval

> **Parameters**
>
> > • **Id** – Chromosome to query
> >
> > • **Chrom** – The Id of the resource
> >
> > • **StartPos** – Get coverage starting at this position. Default is 1
> >
> > • **EndPos** – Get coverage up to and including this position. Default is StartPos + 1280

:return:CoverageResponse – an instance of CoverageResponse

getProjectById(*Id*)
    Request a project object by Id

        Parameters **Id** – The Id of the project

getProjectByUser(*Id*, *queryPars={'Limit': '100', 'SortBy': 'Id', 'SortDir': 'Asc', 'Offset': '0'}*)
    Returns a list available projects for a User with the specified Id

        Parameters

            • **Id** – The id of the user

            • **qp** – An (optional) object of type QueryParameters for custom sorting and filtering

getSampleById(*Id*)
    Returns a Sample object

        Parameters **Id** – The id of the sample

getSamplesByProject(*Id*, *queryPars={'Limit': '100', 'SortBy': 'Id', 'SortDir': 'Asc', 'Offset': '0'}*)
    Returns a list of samples associated with a project with Id

        Parameters

            • **Id** – The id of the project

            • **queryPars** – An (optional) object of type QueryParameters for custom sorting and filtering

getServerUri()
    Returns the server uri used by this instance

getUserById(*Id*)
    Returns the User object corresponding to Id

        Parameters **Id** – The Id of the user

getVariantMetadata(*Id*, *Format*)
    Returns a VariantMetadata object for the variant file

        Parameters

            • **Id** – The Id of the VCF file

            • **Format** – Set to 'vcf' to get the results as lines in VCF format

getVerificationCode(*scope*)
    Returns the BaseSpace dictionary containing the verification code and verification url for the user to approve access to a specific data scope.

    Corresponding curl call: curlCall = 'curl -d "response_type=device_code" -d "client_id=' + client_key + '"' -d "scope=' + scope + '"' ' + deviceURL

    For details see: https://developer.basespace.illumina.com/docs/content/documentation/authentication/obtaining-access-tokens

        Parameters **scope** – The scope that access is requested for

getWebVerificationCode(*scope*, *redirectURL*, *state=''*)
    Generates the URL the user should be redirected to for web-based authentication

        Parameters

            • **scope** – The scope that access is requested for

            • **redirectURL** – The redirect URL

        State An optional state parameter that will passed through to the redirect response

obtainAccessToken (*deviceCode*)
>   Returns a user specific access token.

>> **Parameters deviceCode** – The device code returned by the verification code method

setAppSessionState (*Id*, *Status*, *Summary*)
>   Set the status of an AppResult object

>> **Parameters**

>>> • **Id** – The id of the AppResult

>>> • **Status** – The status assignment string must

>>> • **Summary** – The summary string

setTimeout (*time*)
>   Specify the timeout in seconds for each request made

>> **Parameters time** – timeout in second

1.2 Models

1.2.1 Project

class `BaseSpacePy.model.Project.`**Project**
>   Represents a BaseSpace Project object.

createAppResult (*api*, *name*, *desc*, *appSessionId=None*, *samples=*[ ])
>   Return a newly created app result object

>> **Parameters**

>>> • **api** – An instance of BaseSpaceAPI

>>> • **name** – The name of the app result

>>> • **desc** – A describtion of the app result

getAccessStr (*scope='write'*)
>   Returns the scope-string to used for requesting BaseSpace access to the object

>> **Parameters scope** – The scope-type that is request (write|read)

getAppResults (*api*, *myQp={}*, *statuses=*[ ])
>   Returns a list of AppResult objects.

>> **Parameters**

>>> • **api** – An instance of BaseSpaceAPI

>>> • **statuses** – An optional list of statuses

getSamples (*api*)
>   Returns a list of Sample objects.

>> **Parameters api** – An instance of BaseSpaceAPI

isInit ()
>   Is called to test if the Project instance has been initialized.

>   **Throws:** ModelNotInitializedException - Indicates the object has not been populated yet.

1.2.2 AppSession and AppResult

class `BaseSpacePy.model.AppSession.`**`AppSession`**

AppLaunch contains the data returned

`setStatus` (*api*, *Status*, *Summary*)

Sets the status of the AppSession (note: once set to 'completed' or 'aborted' no more work can be done to the instance)

Parameters

- **api** – An instance of BaseSpaceAPI
- **Status** – The status value, must be completed, aborted, working, or suspended
- **Summary** – The status summary

class `BaseSpacePy.model.AppResult.`**`AppResult`**

`getAccessStr` (*scope='write'*)

Returns the scope-string to be used for requesting BaseSpace access to the object

Parameters **scope** – The scope-type that is request (write|read)

`getFiles` (*api*, *myQp={}*)

Returns a list of file objects

Parameters

- **api** – An instance of BaseSpaceAPI
- **myQp** – (Optional) QueryParameters for sorting and filtering the file list

`getReferencedSamples` (*api*)

Returns a list of sample objects references by the AppResult. NOTE this method makes one request to REST server per sample

`getReferencedSamplesIds` ()

Return a list of sample ids for the samples referenced.

`isInit` ()

Is called to test if the Project instance has been initialized

Throws: ModelNotInitializedException - if the instance has not been populated.

`uploadFile` (*api*, *localPath*, *fileName*, *directory*, *contentType*)

Uploads a local file to the BaseSpace AppResult

Parameters

- **api** – An instance of BaseSpaceAPI
- **localPath** – The local path of the file
- **fileName** – The filename
- **directory** – The remote directory to upload to
- **contentType** – The content-type of the file

1.2.3 Sample

class `BaseSpacePy.model.Sample.`**`Sample`**

Representation of a BaseSpace Sample object.

getAccessStr (*scope='write'*)
    Returns the scope-string to used for requesting BaseSpace access to the sample.

        Parameters scope – The scope type that is request (write|read).

getFiles (*api*, *myQp={}*)
    Returns a list of File objects

        Parameters

- **api** – A BaseSpaceAPI instance

- **myQp** – Query parameters to sort and filter the file list by.

getReferencedAppResults (*api*)
    Return the AppResults referenced by this sample. Note the returned AppResult objects do not have their "References" field set, to get a fully populate AppResult object you must use getAppResultById in BaseSpaceAPI.

isInit ()
    Is called to test if the sample instance has been initialized.

        Throws: ModelNotInitializedException - Indicated the Id variable is not set.

1.2.4 File

class `BaseSpacePy.model.File.`**File**
    Represents a BaseSpace file object.

downloadFile (*api*, *localDir*, *range=*$\big[\,\big]$)
    Download the file object to the specified localDir or a byte range of the file, by specifying the start and stop byte in the range.

        Parameters

- **api** – A BaseSpaceAPI with read access on the scope including the file object.

- **loadlDir** – The local directory to place the file in.

- **range** – Specify the start and stop byte of the file chunk that needs retrieved.

filterVariant (*api*, *Chrom*, *StartPos*, *EndPos*, *q=None*)
    Returns a list of Variant objects available in the specified region

        Parameters

- **api** – An instance of BaseSpaceAPI

- **Chrom** – Chromosome as a string - for example 'chr2'

- **StartPos** – The start position of region of interest as a string

- **EndPos** – The end position of region of interest as a string

- **q** – An instance of

getCoverageMeta (*api*, *Chrom*)
    Return an object of CoverageMetadata for the selected region

        Parameters

- **api** – An instance of BaseSpaceAPI.

- **Chrom** – The chromosome of interest.

getIntervalCoverage(*api*, *Chrom*, *StartPos*, *EndPos*)
    Return a coverage object for the specified region and chromosome.

> **Parameters**
>
> - **api** – An instance of BaseSpaceAPI
> - **Chrom** – Chromosome as a string - for example 'chr2'
> - **StartPos** – The start position of region of interest as a string
> - **EndPos** – The end position of region of interest as a string

getVariantMeta(*api*)
    Return the the meta info for a VCF file as a VariantInfo object

> **Parameters api** – An instance of BaseSpaceAPI

isInit()
    Is called to test if the File instance has been initialized.

> **Throws:** ModelNotInitializedException if the instance has not been populated yet.

isValidFileOption(*filetype*)
    Is called to test if the File instance is matches the filtype parameter

> **Parameters filetype** – The filetype for coverage or variant requests

1.2.5 QueryParameters

class BaseSpacePy.model.QueryParameters.**QueryParameters**(*pars={}, required=['SortBy', 'Offset', 'Limit', 'SortDir']*)
    The QueryParameters class can be passed as an optional arguments for a specific sorting of list-responses (such as lists of sample, AppResult, or variants)

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

INDEX

I

O

P

Q

S

U