
B2_Python_SDK

Release 0.1.8

Backblaze

Oct 16, 2019

CONTENTS

1	Installation Guide	3
1.1	Installing b2sdk	3
1.2	Developer Info	3
2	Quick start guide	5
2.1	Initialize API	5
2.2	Account authorization	5
2.3	Synchronization	5
2.4	Bucket actions	6
2.5	File actions	7
2.6	Inspect account info	10
3	API types	13
3.1	Public interface	13
3.2	Internal interface	13
4	API Reference	15
4.1	Public API	15
4.2	Internal API	57
5	Indices and tables	71
	Python Module Index	73
	Index	75

b2sdk is a client library for easy access to all of the capabilities of B2 Cloud Storage.

B2 command-line tool is an example of how it can be used to provide command-line access to the B2 service, but there are many possible applications (including FUSE filesystems, storage backend drivers for backup applications etc).

INSTALLATION GUIDE

1.1 Installing b2sdk

1.1.1 Installing as a dependency

b2sdk can simply be added to `requirements.txt` (or equivalent such as `setup.py`, `.pipfile`). In order to properly set a dependency, see [this](#) chapter for details.

Caution: The stability of your application depends on correct pinning of versions

1.1.2 Installing for development

b2sdk runs on Python 2.7+ under CPython and PyPy.

To install **b2sdk**, simply run:

```
pip install b2sdk
```

in your python environment.

Note: If you see a message saying that the `six` library cannot be installed, which happens if you're installing with the system python on OS X El Capitan, try this:

```
pip install --ignore-installed b2sdk
```

1.2 Developer Info

We encourage outside contributors to perform changes on our codebase. Many such changes have been merged already. In order to make it easier to contribute, core developers of this project:

- provide guidance (through the issue reporting system)
- provide tool assisted code review (through the Pull Request system)
- maintain a set of integration tests (run with a production cloud)
- maintain a set of (well over a hundred) unit tests

- automatically run unit tests on 14 versions of python (including osx, Jython and pypy)
- format the code automatically using `yapf`
- use static code analysis to find subtle/potential issues with maintainability
- maintain other Continuous Integration tools (coverage tracker)

You'll need to some Python packages installed. To get all the latest things:

```
pip install --upgrade --upgrade-strategy eager -r requirements.txt -r requirements-  
test.txt -r requirements-setup.txt
```

There is a *Makefile* with a rule to run the unit tests using the currently active Python:

```
make setup  
make test
```

will install the required packages, then run the unit tests.

To test in multiple python virtual environments, set the environment variable `PYTHON_VIRTUAL_ENVS` to be a space-separated list of their root directories. When set, the makefile will run the unit tests in each of the environments.

Before checking in, use the `pre-commit.sh` script to check code formatting, run unit tests, run integration tests etc.

The integration tests need a file in your home directory called `.b2_auth` that contains two lines with nothing on them but your account ID and application key:

```
accountId  
applicationKey
```

We marked the places in the code which are significantly less intuitive than others in a special way. To find them occurrences, use `git grep '*magic*'`.

QUICK START GUIDE

2.1 Initialize API

```
>>> from b2sdk.account_info.in_memory import InMemoryAccountInfo
>>> from b2sdk.account_info.sqlite_account_info import SqliteAccountInfo
>>> from b2sdk.api import B2Api

>>> info = InMemoryAccountInfo() # to store credentials, tokens and cache in memory_
↪ OR
>>> info = SqliteAccountInfo() # to store credentials, tokens and cache in ~/.b2_
↪ account_info
>>> b2_api = B2Api(info)
```

To find out more about API object initialization, see `b2sdk.api.B2Api.__init__()`.

2.2 Account authorization

```
>>> application_key_id = '4a5b6c7d8e9f' # get credentials from from B2 website
>>> application_key = '001b8e23c26ff6efb941e237deb182b9599a84bef7'
>>> b2_api.authorize_account("production", application_key_id, application_key)
```

To find out more about account authorization, see `b2sdk.api.B2Api.authorize_account()`

2.3 Synchronization

```
>>> from b2sdk.sync.scan_policies import ScanPoliciesManager
>>> from b2sdk.sync import parse_sync_folder, sync_folders
>>> import time
>>> import sys

>>> source = '/home/user1/b2_example'
>>> destination = 'b2://example-mybucket-b2'

>>> source = parse_sync_folder(source, b2_api)
>>> destination = parse_sync_folder(destination, b2_api)

>>> policies_manager = ScanPoliciesManager(exclude_all_symlinks=True)
```

(continues on next page)

(continued from previous page)

```
>>> sync_folders(
    source_folder=source,
    dest_folder=destination,
    args=args,
    now_millis=int(round(time.time() * 1000)),
    stdout=sys.stdout,
    no_progress=False,
    max_workers=10,
    policies_manager=policies_manager,
    dry_run=False,
    allow_empty_source=True,
)
upload some.pdf
upload som2.pdf
```

Tip: Sync is the preferred way of getting data into and out of B2 cloud, because it can achieve *highest performance* due to parallelization of scanning and data transfer operations.

To learn more about sync, see [Sync](#).

2.4 Bucket actions

2.4.1 List buckets

```
>>> b2_api.list_buckets()
[Bucket<346501784642eb3e60980d10,example-mybucket-b2-1,allPublic>]
>>> for b in b2_api.list_buckets():
    print('%s %-10s %s' % (b.id_, b.type_, b.name))
346501784642eb3e60980d10 allPublic example-mybucket-b2-1
```

2.4.2 Create a bucket

```
>>> bucket_name = 'example-mybucket-b2-1' # must be unique in B2 (across all_
↳accounts!)
>>> bucket_type = 'allPublic' # or 'allPrivate'

>>> b2_api.create_bucket(bucket_name, bucket_type)
Bucket<346501784642eb3e60980d10,example-mybucket-b2-1,allPublic>
```

You can optionally store bucket info, CORS rules and lifecycle rules with the bucket. See `b2sdk.api.create_bucket()`.

2.4.3 Remove a bucket

```
>>> bucket_name = 'example-mybucket-b2-to-delete'
>>> bucket = b2_api.get_bucket_by_name(bucket_name)
>>> b2_api.delete_bucket(bucket)
{'accountId': '451862be08d0',
```

(continues on next page)

(continued from previous page)

```
'bucketId': '346501784642eb3e60980d10',
'bucketInfo': {},
'bucketName': 'example-mybucket-b2-to-delete',
'bucketType': 'allPublic',
'corsRules': [],
'lifecycleRules': [],
'revision': 3}
```

2.4.4 Update bucket info

```
>>> new_bucket_type = 'allPrivate'
>>> bucket_name = 'example-mybucket-b2'

>>> bucket = b2sdk.api.get_bucket_by_name(bucket_name)
>>> bucket.update(bucket_type=new_bucket_type)
{'accountId': '451862be08d0',
 'bucketId': '5485a1682662eb3e60980d10',
 'bucketInfo': {},
 'bucketName': 'example-mybucket-b2',
 'bucketType': 'allPrivate',
 'corsRules': [],
 'lifecycleRules': [],
 'revision': 3}
```

For more information see `b2sdk.bucket.Bucket.update()`.

2.5 File actions

Tip: Sync is the preferred way of getting files into and out of B2 cloud, because it can achieve *highest performance* due to parallelization of scanning and data transfer operations.

To learn more about sync, see [Sync](#).

Use the functions described below only if you *really* need to transfer a single file.

2.5.1 Upload file

```
>>> from b2sdk.progress import make_progress_listener

>>> local_file_path = '/home/user1/b2_example/new.pdf'
>>> b2_file_name = 'dummy_new.pdf'
>>> file_info = {'how': 'good-file'}

>>> bucket = b2sdk.api.get_bucket_by_name(bucket_name)
>>> bucket.upload_local_file(
    local_file=local_file_path,
    file_name=b2_file_name,
    file_infos=file_info,
)
<b2sdk.file_version.FileVersionInfo at 0x7fc8cd560550>
```

2.5.2 Download file

By id

```
>>> from b2sdk.progress import make_progress_listener
>>> from b2sdk.download_dest import DownloadDestLocalFile

>>> local_file_path = '/home/user1/b2_example/new2.pdf'
>>> file_id = '4_z5485a1682662eb3e60980d10_f1195145f42952533_d20190403_m130258_c002_
↳v0001111_t0002'
>>> progress_listener = make_progress_listener(local_file_path, True)
>>> download_dest = DownloadDestLocalFile(local_file_path)
>>> b2_api.download_file_by_id(file_id, download_dest, progress_listener)
{'fileId': '4_z5485a1682662eb3e60980d10_f1195145f42952533_d20190403_m130258_c002_
↳v0001111_t0002',
 'fileName': 'som2.pdf',
 'contentType': 'application/pdf',
 'contentLength': 1870579,
 'contentShal': 'd821849a70922e87c2b0786c0be7266b89d87df0',
 'fileInfo': {'src_last_modified_millis': '1550988084299'}}

>>> print('File name: ', download_dest.file_name)
File name: som2.pdf
>>> print('File id: ', download_dest.file_id)
File id: 4_z5485a1682662eb3e60980d10_f1195145f42952533_d20190403_m130258_c002_
↳v0001111_t0002
>>> print('File size: ', download_dest.content_length)
File size: 1870579
>>> print('Content type:', download_dest.content_type)
Content type: application/pdf
>>> print('Content shal:', download_dest.content_shal)
Content shal: d821849a70922e87c2b0786c0be7266b89d87df0
```

By name

```
>>> bucket = b2_api.get_bucket_by_name(bucket_name)
>>> b2_file_name = 'dummy_new.pdf'
>>> local_file_name = '/home/user1/b2_example/new3.pdf'
>>> download_dest = DownloadDestLocalFile(local_file_name)
>>> progress_listener = make_progress_listener(local_file_path, True)
>>> bucket.download_file_by_name(b2_file_name, download_dest, progress_listener)
{'fileId': '4_z5485a1682662eb3e60980d10_f113f963288e711a6_d20190404_m065910_c002_
↳v0001095_t0044',
 'fileName': 'dummy_new.pdf',
 'contentType': 'application/pdf',
 'contentLength': 1870579,
 'contentShal': 'd821849a70922e87c2b0786c0be7266b89d87df0',
 'fileInfo': {'how': 'good-file'}}
```

2.5.3 List files

```
>>> bucket_name = 'example-mybucket-b2'
>>> bucket = b2_api.get_bucket_by_name(bucket_name)
```

(continues on next page)

(continued from previous page)

```

>>> max_to_show = 1 # max files to show, default=100, optional parameter
>>> start_file_name = 'som' # default is '', optional parameter
>>> bucket.list_file_names(start_file_name, max_to_show)
{'files': [{'accountId': '451862be08d0',
  'action': 'upload',
  'bucketId': '5485a1682662eb3e60980d10',
  'contentLength': 1870579,
  'contentShal': 'd821849a70922e87c2b0786c0be7266b89d87df0',
  'contentType': 'application/pdf',
  'fileId': '4_z5485a1682662eb3e60980d10_f1195145f42952533_d20190403_m130258_c002_
↪v0001111_t0002',
  'fileInfo': {'src_last_modified_millis': '1550988084299'},
  'fileName': 'som2.pdf',
  'uploadTimestamp': 1554296578000}],
'nextFileName': 'som2.pdf '}

# list file versions
>>> bucket.list_file_versions()
{'files': [{'accountId': '451862be08d0',
  'action': 'upload',
  'bucketId': '5485a1682662eb3e60980d10',
  'contentLength': 1870579,
  'contentShal': 'd821849a70922e87c2b0786c0be7266b89d87df0',
  'contentType': 'application/pdf',
  'fileId': '4_z5485a1682662eb3e60980d10_f1195145f42952533_d20190403_m130258_c002_
↪v0001111_t0002',
  'fileInfo': {'src_last_modified_millis': '1550988084299'},
  'fileName': 'som2.pdf',
  'uploadTimestamp': 1554296578000}

```

2.5.4 Get file meta information

```

>>> file_id = '4_z5485a1682662eb3e60980d10_f113f963288e711a6_d20190404_m065910_c002_
↪v0001095_t0044'
>>> b2_api.get_file_info(file_id)
{'accountId': '451862be08d0',
  'action': 'upload',
  'bucketId': '5485a1682662eb3e60980d10',
  'contentLength': 1870579,
  'contentShal': 'd821849a70922e87c2b0786c0be7266b89d87df0',
  'contentType': 'application/pdf',
  'fileId': '4_z5485a1682662eb3e60980d10_f113f963288e711a6_d20190404_m065910_c002_
↪v0001095_t0044',
  'fileInfo': {'how': 'good-file'},
  'fileName': 'dummy_new.pdf',
  'uploadTimestamp': 1554361150000}

```

2.5.5 Copy file

```

>>> file_id = '4_z5485a1682662eb3e60980d10_f118df9ba2c5131e8_d20190619_m065809_c002_
↪v0001126_t0040'
>>> bucket.copy_file(file_id, 'f2_copy.txt')

```

(continues on next page)

(continued from previous page)

```
{'accountId': '451862be08d0',
 'action': 'copy',
 'bucketId': '5485a1682662eb3e60980d10',
 'contentLength': 124,
 'contentSha1': '737637702a0e41dda8b7be79c8db1d369c6eef4a',
 'contentType': 'text/plain',
 'fileId': '4_z5485a1682662eb3e60980d10_f1022e2320daf707f_d20190620_m122848_c002_
↪v0001123_t0020',
 'fileInfo': {'src_last_modified_millis': '1560848707000'},
 'fileName': 'f2_copy.txt',
 'uploadTimestamp': 1561033728000}
```

If you want to copy just the part of the file, then you can specify the `bytes_range` as a tuple.

```
>>> file_id = '4_z5485a1682662eb3e60980d10_f118df9ba2c5131e8_d20190619_m065809_c002_
↪v0001126_t0040'
>>> bucket.copy_file(file_id, 'f2_copy.txt', bytes_range=(8,15))
{'accountId': '451862be08d0',
 'action': 'copy',
 'bucketId': '5485a1682662eb3e60980d10',
 'contentLength': 8,
 'contentSha1': '274713be564aeca8e8de362acb68658b576d0b40',
 'contentType': 'text/plain',
 'fileId': '4_z5485a1682662eb3e60980d10_f114b0c11b6b6e39e_d20190620_m122007_c002_
↪v0001123_t0004',
 'fileInfo': {'src_last_modified_millis': '1560848707000'},
 'fileName': 'f2_copy.txt',
 'uploadTimestamp': 1561033207000}
```

For more information see `b2sdk.v1.Bucket.copy_file()`.

2.5.6 Delete file

```
>>> file_id = '4_z5485a1682662eb3e60980d10_f113f963288e711a6_d20190404_m065910_c002_
↪v0001095_t0044'
>>> file_info = b2_api.delete_file_version(file_id, 'dummy_new.pdf')
>>>
```

2.5.7 Cancel file operations

```
>>> bucket = b2_api.get_bucket_by_name(bucket_name)
>>> for file_version in bucket.list_unfinished_large_files():
    bucket.cancel_large_file(file_version.file_id)
>>>
```

2.6 Inspect account info

```
TODO

account_info = b2_api.account_info
```

(continues on next page)

(continued from previous page)

```
accountId = account_info.get_account_id()
allowed = account_info.get_allowed()
applicationKey = account_info.get_application_key()
accountAuthToken = account_info.get_account_auth_token()
apiUrl = account_info.get_api_url()
downloadUrl = account_info.get_download_url()
```


API TYPES

b2sdk API is divided into two parts, *public* and *internal*. Please pay attention to which interface type you use and pin the version range accordingly.

Caution: The stability of your application depends on correct pinning of versions.

3.1 Public interface

Public interface consists of **public** members of modules listed in *API Public* section

Those modules will not change in a backwards-incompatible way between non-major versions.

This should be used in 99% of use cases, it's enough to implement anything from a web application to a [FUSE filesystem](#).

Hint: If the current version of **b2sdk** is 4.5.6 and you only use the *public* interface, put this in your `requirements.txt` to be safe:

```
b2sdk>=4.5.6,<5.0.0
```

Note: `b2sdk.*._something` and `b2sdk.*.*._something`, having a name which begins with an underscore, are NOT considered public interface.

3.2 Internal interface

Some rarely used features of B2 cloud are not implemented in **b2sdk**. Tracking usage of transactions and transferred data is a good example - if it is required, additional work would need to be put into a specialized internal interface layer to enable tracking and reporting.

b2sdk maintainers are very supportive in case someone wants to contribute an additional feature. Please consider adding it to the sdk, so that it's centrally supported (unlike your own implementation of your own, which would not receive updates).

Internal interface modules are listed in *API Internal* section.

Note: It is ok for you to use our internal interface (better that, than copying our source files!), however if you do, please pin your dependencies to **middle** version as backwards-incompatible changes may be introduced in a non-major version.

Hint: If the current version of **b2sdk** is 4.5.6 and you use the *internal* interface, put this in your requirements.txt:

`b2sdk>=4.5.6,<4.6.0`

API REFERENCE

4.1 Public API

Hint: Use *Quick start guide* to quickly jump into examples

4.1.1 Authentication

`b2sdk.account_info.abstract` – Account info abstract

class `b2sdk.account_info.abstract.AbstractAccountInfo`

Bases: `object`

Holder for all account-related information that needs to be kept between API calls, and between invocations of the command-line tool. This includes: account ID, application key ID, application key, auth tokens, API URL, download URL, and uploads URLs.

This class must be THREAD SAFE because it may be used by multiple threads running in the same Python process. It also needs to be safe against multiple processes running at the same time.

`DEFAULT_ALLOWED = {'bucketId': None, 'bucketName': None, 'capabilities': ['listKeys`

`REALM_URLS = {'dev': 'http://api.backblazeb2.xyz:8180', 'production': 'https://api.b`

classmethod `all_capabilities()`

Return a list of all possible capabilities

Return type `list`

classmethod `allowed_is_valid(allowed)`

Makes sure that all of the required fields are present, and that `bucketId` is set if `bucketName` is.

If the `bucketId` is for a bucket that no longer exists, or the capabilities do not allow `listBuckets`, then we won't have a `bucketName`.

Parameters `allowed(dict)` – the structure to use for old account info that was saved without 'allowed'

Return type `bool`

abstract `clear()`

Removes all stored information

abstract `clear_bucket_upload_data(bucket_id)`

Removes all upload URLs for the given bucket.

Parameters `bucket_id` (*str*) – a bucket ID

abstract `clear_large_file_upload_urls` (*file_id*)

Clear a pool of URLs for a given file ID

Parameters `file_id` (*str*) – a file ID

abstract `get_account_auth_token` ()

Returns `account_auth_token` or raises `MissingAccountData` exception

Return type *str*

abstract `get_account_id` ()

Returns account ID or raises `MissingAccountData` exception

Return type *str*

`get_account_id_or_app_key_id` ()

Returns the application key ID used to authenticate

Return type *str*

Deprecated since version 0.1.6: Use `get_application_key_id` () instead.

abstract `get_allowed` ()

An ‘allowed’ dict, as returned by `b2_authorize_account`. Never `None`; for account info that was saved before ‘allowed’ existed, returns `DEFAULT_ALLOWED`.

Return type *dict*

abstract `get_api_url` ()

Returns `api_url` or raises `MissingAccountData` exception

Return type *str*

abstract `get_application_key` ()

Returns `application_key` or raises `MissingAccountData` exception

Return type *str*

abstract `get_application_key_id` ()

Returns the application key ID used to authenticate

Return type *str*

abstract `get_bucket_id_or_none_from_bucket_name` (*bucket_name*)

Looks up the bucket ID for a given bucket name.

Parameters `bucket_name` (*str*) – a bucket name

Return bucket ID or None

Return type *str, None*

abstract `get_download_url` ()

Returns `download_url` or raises `MissingAccountData` exception

Return type *str*

abstract `get_minimum_part_size` ()

Return the minimum number of bytes in a part of a large file

Returns number of bytes

Return type *int*

abstract `get_realm()`

Returns realm or raises MissingAccountData exception

Return type `str`

abstract `put_bucket_upload_url(bucket_id, upload_url, upload_auth_token)`

Add an (upload_url, upload_auth_token) pair to the pool available for the bucket.

Parameters

- **bucket_id** (`str`) – a bucket ID
- **upload_url** (`str`) – an upload URL
- **upload_auth_token** (`str`) – an upload authentication token

Return type `tuple`

abstract `put_large_file_upload_url(file_id, upload_url, upload_auth_token)`

Put large file upload URL into a pool

Parameters

- **file_id** (`str`) – a file ID
- **upload_url** (`str`) – an upload URL
- **upload_auth_token** (`str`) – an upload authentication token

abstract `refresh_entire_bucket_name_cache(name_id_iterable)`

Removes all previous name-to-id mappings and stores new ones.

Parameters **name_id_iterable** (`list`) – a list of tuples of the form (name, id)

abstract `remove_bucket_name(bucket_name)`

Removes one entry from the bucket name cache.

Parameters **bucket_name** (`str`) – a bucket name

abstract `save_bucket(bucket)`

Remembers the ID for a bucket name.

Parameters **bucket** (`b2sdk.bucket.Bucket`) – a Bucket object

set_auth_data (`account_id, auth_token, api_url, download_url, minimum_part_size, application_key, realm, allowed=None, application_key_id=None`)

Stores the results of `b2_authorize_account`.

All of the information returned by `b2_authorize_account` is saved, because all of it is needed by some command.

The allowed structure is the one returned `b2_authorize_account`, with the addition of a bucket-Name field. For keys with bucket restrictions, the name of the bucket is looked up and stored, too. The console_tool does everything by bucket name, so it's convenient to have the restricted bucket name handy.

Parameters

- **account_id** (`str`) – user account ID
- **auth_token** (`str`) – user authentication token
- **api_url** (`str`) – an API URL
- **download_url** (`str`) – path download URL
- **minimum_part_size** (`int`) – minimum size of the file part
- **application_key** (`str`) – application key

- **realm** (*str*) – a realm to authorize account in
- **allowed** (*dict*) – the structure to use for old account info that was saved without ‘allowed’
- **application_key_id** (*str*) – application key ID

Changed in version 0.1.5: *account_id_or_app_key_id* renamed to *get_application_key_id*

abstract take_bucket_upload_url (*bucket_id*)

Returns a pair (upload_url, upload_auth_token) that has been removed from the pool for this bucket, or (None, None) if there are no more left.

Parameters **bucket_id** (*str*) – a bucket ID

Return type *tuple*

abstract take_large_file_upload_url (*file_id*)

Take large file upload URL from a pool

Parameters **file_id** (*str*) – a file ID

b2sdk.account_info.exception – Account info exceptions

exception `b2sdk.account_info.exception.AccountInfoError` (*args, **kwargs)

Bases: `b2sdk.exception.B2Error`

Base class for all account info errors

exception `b2sdk.account_info.exception.CorruptAccountInfo` (*file_name*)

Bases: `b2sdk.account_info.exception.AccountInfoError`

Raised when an account info file is corrupted

__init__ (*file_name*)

Parameters **file_name** (*str*) – an account info file name

exception `b2sdk.account_info.exception.MissingAccountData` (*key*)

Bases: `b2sdk.account_info.exception.AccountInfoError`

Raised when there is no account info data available

__init__ (*key*)

Parameters **key** (*str*) – a key for getting account data

b2sdk.account_info.sqlite_account_info – SQLite account info

AccountInfo class which uses a [SQLite database](#) as backend.

The database has a following schema:

account					
o account_auth_token	TEXT				
o account_id	TEXT				
o account_id_or_app_key_id	TEXT				
o allowed	TEXT				
o api_url	TEXT				
o application_key	TEXT				
o download_url	TEXT				
o minimum_part_size	INTEGER				
o realm	TEXT				

bucket					
o bucket_id	TEXT				
o bucket_name	TEXT				

bucket_upload_url					
o bucket_id	TEXT				
o upload_auth_token	TEXT				
o upload_url	TEXT				

update_done					
o update_number	INTEGER				

generated by sadisplay v0.4.9

class `b2sdk.account_info.sqlite_account_info.SqliteAccountInfo` (*file_name=None*,
last_upgrade_to_run=None)

Bases: `b2sdk.account_info.upload_url_pool.UrlPoolAccountInfo`

Stores account information in an sqlite database, which is used to manage concurrent access to the data.

The 'update_done' table tracks the schema updates that have been completed.

__init__ (*file_name=None*, *last_upgrade_to_run=None*)

Parameters

- **file_name** (*str*) – The sqlite file to use; overrides the default.
- **last_upgrade_to_run** (*int*) – For testing only, override the auto-update on the db.

clear ()

Remove all info about accounts and buckets

get_account_auth_token ()

Return account authentication token

Return type *str*

get_account_id ()

Return account ID

Return type *str*

get_allowed ()

Return 'allowed' dictionary info. The 'allowed' column was not in the original schema, so it may be NULL.

Return type *dict*

get_api_url ()

Return API URL

Return type *str*

get_application_key ()

Return application key

Return type *str*

get_application_key_id ()

Returns an application key ID. The 'account_id_or_app_key_id' column was not in the original schema, so it may be NULL.

In addition, this is the only place where we are not renaming `account_id_or_app_key_id` to `application_key_id` because it requires a column change.

`application_key_id == account_id_or_app_key_id`

Return type `str`

`get_bucket_id_or_none_from_bucket_name` (*bucket_name*)

Return bucket ID or None by a given name

Parameters `bucket_name` (*str*) – a bucket name

Return type `str, None`

`get_download_url` ()

Return download URL

Return type `str`

`get_minimum_part_size` ()

Return minimum part size

Return type `int`

`get_realm` ()

Return realm name

Return type `str`

`refresh_entire_bucket_name_cache` (*name_id_iterable*)

Refresh names and IDs of buckets

Parameters `name_id_iterable` – an iterable which yields bucket name and ID

`remove_bucket_name` (*bucket_name*)

Remove bucket info by a given name

Parameters `bucket_name` (*str*) – a bucket name

`save_bucket` (*bucket*)

Save bucket info

Parameters `bucket` – a Bucket object

Type *b2sdk.bucket.Bucket*

`set_auth_data_with_schema_0_for_test` (*account_id, auth_token, api_url, download_url, minimum_part_size, application_key, realm*)

Set authentication data for tests

Parameters

- **`account_id`** (*str*) – an account ID
- **`auth_token`** (*str*) – an authentication token
- **`api_url`** (*str*) – an API URL
- **`download_url`** (*str*) – a download URL
- **`minimum_part_size`** (*int*) – a minimum part size
- **`application_key`** (*str*) – an application key
- **`realm`** (*str*) – a realm to authorize account in

b2sdk.account_info.upload_url_pool – Upload URL pool**class** b2sdk.account_info.upload_url_pool.UploadUrlPoolBases: `object`

For each key (either a bucket id or large file id), holds a pool of (url, auth_token) pairs, with thread-safe methods to add and remove them.

__init__ ()

Initialize self. See help(type(self)) for accurate signature.

clear_for_key (*key*)

Remove an item from the pool by key

Parameters **key** – bucket ID or large file ID**put** (*key*, *url*, *auth_token*)

Adds the url and auth token to the pool for the given key.

Parameters

- **key** – bucket ID or large file ID
- **url** (*str*) – bucket or file URL
- **auth_token** (*str*) – authentication token

take (*key*)

Returns (url, auth_token) if one is available, or (None, None) if not.

Parameters **key** – bucket ID or large file ID**Return type** `tuple`**class** b2sdk.account_info.upload_url_pool.UrlPoolAccountInfoBases: `b2sdk.account_info.abstract.AbstractAccountInfo`

Holder for all account-related information that needs to be kept between API calls, and between invocations of the command-line tool. This includes: account ID, application key, auth tokens, API URL, download URL, and uploads URLs.

This concrete implementation uses an instance of UploadUrlPool as an underlying storage

__init__ ()

Initialize self. See help(type(self)) for accurate signature.

abstract clear ()

Remove all stored information

clear_bucket_upload_data (*bucket_id*)

Removes all upload URLs for the given bucket.

Parameters **bucket_id** (*str*) – a bucket ID**clear_large_file_upload_urls** (*file_id*)

Clear a pool of URLs for a given file ID

Parameters **file_id** (*str*) – a file ID**put_bucket_upload_url** (*bucket_id*, *upload_url*, *upload_auth_token*)

Add an (upload_url, upload_auth_token) pair to the pool available for the bucket.

Parameters

- **bucket_id** (*str*) – a bucket ID

- **upload_url** (*str*) – an upload URL
- **upload_auth_token** (*str*) – an upload authentication token

Return type *tuple*

put_large_file_upload_url (*file_id*, *upload_url*, *upload_auth_token*)

Put large file upload URL into a pool

Parameters

- **file_id** (*str*) – a file ID
- **upload_url** (*str*) – an upload URL
- **upload_auth_token** (*str*) – an upload authentication token

take_bucket_upload_url (*bucket_id*)

Returns a pair (upload_url, upload_auth_token) that has been removed from the pool for this bucket, or (None, None) if there are no more left.

Parameters **bucket_id** (*str*) – a bucket ID

Return type *tuple*

take_large_file_upload_url (*file_id*)

Take large file upload URL from a pool

Parameters **file_id** (*str*) – a file ID

4.1.2 Sources and Destinations

b2sdk.download_dest – Download destination

class `b2sdk.download_dest.AbstractDownloadDestination`

Bases: `object`

Interface to a destination for a downloaded file.

abstract make_file_context (*file_id*, *file_name*, *content_length*, *content_type*, *content_shal*,
file_info, *mod_time_millis*, *range_=None*)

Returns a context manager that yields a binary file-like object to use for writing the contents of the file.

Parameters

- **file_id** – the B2 file ID from the headers
- **file_name** – the B2 file name from the headers
- **content_type** – the content type from the headers
- **content_shal** – the content sha1 from the headers (or “none” for large files)
- **file_info** – the user file info from the headers
- **mod_time_millis** – the desired file modification date in ms since 1970-01-01
- **range** – starting and ending offsets of the received file contents. Usually None, which means that the whole file is downloaded.

Returns None

class `b2sdk.download_dest.DownloadDestBytes`

Bases: `b2sdk.download_dest.AbstractDownloadDestination`

Stores a downloaded file into bytes in memory.

__init__()

Initialize self. See help(type(self)) for accurate signature.

capture_bytes_context()

Remembers the bytes written in self.bytes_written

get_bytes_written()

make_file_context(*file_id, file_name, content_length, content_type, content_sha1, file_info, mod_time_millis, range_=None*)

Returns a context manager that yields a binary file-like object to use for writing the contents of the file.

Parameters

- **file_id** – the B2 file ID from the headers
- **file_name** – the B2 file name from the headers
- **content_type** – the content type from the headers
- **content_sha1** – the content sha1 from the headers (or “none” for large files)
- **file_info** – the user file info from the headers
- **mod_time_millis** – the desired file modification date in ms since 1970-01-01
- **range** – starting and ending offsets of the received file contents. Usually None, which means that the whole file is downloaded.

Returns None

class b2sdk.download_dest.**DownloadDestLocalFile**(*local_file_path*)

Bases: *b2sdk.download_dest.AbstractDownloadDestination*

Stores a downloaded file into a local file and sets its modification time.

MODE = 'wb+'

__init__(*local_file_path*)

Initialize self. See help(type(self)) for accurate signature.

make_file_context(*file_id, file_name, content_length, content_type, content_sha1, file_info, mod_time_millis, range_=None*)

Returns a context manager that yields a binary file-like object to use for writing the contents of the file.

Parameters

- **file_id** – the B2 file ID from the headers
- **file_name** – the B2 file name from the headers
- **content_type** – the content type from the headers
- **content_sha1** – the content sha1 from the headers (or “none” for large files)
- **file_info** – the user file info from the headers
- **mod_time_millis** – the desired file modification date in ms since 1970-01-01
- **range** – starting and ending offsets of the received file contents. Usually None, which means that the whole file is downloaded.

Returns None

write_to_local_file_context(*mod_time_millis*)

```
class b2sdk.download_dest.DownloadDestProgressWrapper (download_dest,
                                                    progress_listener)
    Bases: b2sdk.download_dest.AbstractDownloadDestination
    Wraps a DownloadDestination, and reports progress to a ProgressListener.
    __init__ (download_dest, progress_listener)
        Initialize self. See help(type(self)) for accurate signature.
    make_file_context (file_id, file_name, content_length, content_type, content_sha1, file_info,
                      mod_time_millis, range_=None)
        Returns a context manager that yields a binary file-like object to use for writing the contents of the file.
```

Parameters

- **file_id** – the B2 file ID from the headers
- **file_name** – the B2 file name from the headers
- **content_type** – the content type from the headers
- **content_sha1** – the content sha1 from the headers (or “none” for large files)
- **file_info** – the user file info from the headers
- **mod_time_millis** – the desired file modification date in ms since 1970-01-01
- **range** – starting and ending offsets of the received file contents. Usually None, which means that the whole file is downloaded.

Returns None

```
write_file_and_report_progress_context (file_id, file_name, content_length, content_type,
                                         content_sha1, file_info, mod_time_millis, range_)
class b2sdk.download_dest.PreSeekedDownloadDest (local_file_path, seek_target)
    Bases: b2sdk.download_dest.DownloadDestLocalFile
    Stores a downloaded file into a local file and sets its modification time. Does not truncate the target file, seeks to a given offset just after opening a descriptor.
    MODE = 'rb+'
    __init__ (local_file_path, seek_target)
        Initialize self. See help(type(self)) for accurate signature.
    write_to_local_file_context (*args, **kwargs)
```

b2sdk.upload_source – Upload source

```
class b2sdk.upload_source.AbstractUploadSource
    Bases: object
    The source of data for uploading to b2.
    abstract get_content_length ()
        Returns the number of bytes of data in the file.
    abstract get_content_sha1 ()
        Return a 40-character string containing the hex SHA1 checksum of the data in the file.
    abstract open ()
        Returns a binary file-like object from which the data can be read. :return:
```

```

class b2sdk.upload_source.UploadSourceBytes (data_bytes)
    Bases: b2sdk.upload_source.AbstractUploadSource

    __init__ (data_bytes)
        Initialize self. See help(type(self)) for accurate signature.

    get_content_length ()
        Returns the number of bytes of data in the file.

    get_content_sha1 ()
        Return a 40-character string containing the hex SHA1 checksum of the data in the file.

    open ()
        Returns a binary file-like object from which the data can be read. :return:

class b2sdk.upload_source.UploadSourceLocalFile (local_path, content_sha1=None)
    Bases: b2sdk.upload_source.AbstractUploadSource

    __init__ (local_path, content_sha1=None)
        Initialize self. See help(type(self)) for accurate signature.

    get_content_length ()
        Returns the number of bytes of data in the file.

    get_content_sha1 ()
        Return a 40-character string containing the hex SHA1 checksum of the data in the file.

    open ()
        Returns a binary file-like object from which the data can be read. :return:

```

4.1.3 b2sdk.api – API client

```

class b2sdk.api.B2Api (account_info=None, cache=None, raw_api=None,
                      max_upload_workers=10)
    Bases: object

```

Provides file-level access to B2 services.

While B2RawApi provides direct access to the B2 web APIs, this class handles several things that simplify the task of uploading and downloading files: - re-acquires authorization tokens when they expire - retrying uploads when an upload URL is busy - breaking large files into parts - emulating a directory structure (B2 buckets are flat)

Adds an object-oriented layer on top of the raw API, so that buckets and files returned are Python objects with accessor methods.

Also, keeps a cache of information needed to access the service, such as auth tokens and upload URLs.

```

__init__ (account_info=None, cache=None, raw_api=None, max_upload_workers=10)
    Initializes the API using the given account info.

```

Parameters

- **account_info** – an instance of *UrlPoolAccountInfo*, or any custom class derived from *AbstractAccountInfo*
- **cache** – an instance of the one of the following classes: *DummyCache*, *InMemoryCache*, *AuthInfoCache*, or any custom class derived from *AbstractCache*
- **raw_api** – an instance of one of the following classes: *B2RawApi*, *RawSimulator*, or any custom class derived from *AbstractRawApi*

- **max_upload_workers** (*int*) – a number of upload threads, default is 10

authorize_account (*realm, application_key_id, application_key*)

Perform account authorization

Parameters

- **realm** (*str*) – a realm to authorize account in (usually just “production”)
- **application_key_id** (*str*) – application key ID
- **application_key** (*str*) – user’s application key

authorize_automatically ()

Perform automatic account authorization, retrieving all account data from account info object passed during initialization

cancel_large_file (*file_id*)

Cancel a large file upload

Parameters **file_id** (*str*) – a file ID

check_bucket_restrictions (*bucket_name*)

Checks to see if the allowed field from authorize-account has a bucket restriction.

If it does, does the bucket_name for a given api call match that. If not it raises a RestrictedBucket error.

Parameters **bucket_name** (*str*) – a bucket name

create_bucket (*name, bucket_type, bucket_info=None, cors_rules=None, lifecycle_rules=None*)

Create a bucket

Parameters

- **name** (*str*) – bucket name
- **bucket_type** (*str*) – a bucket type, could be one of the following values: “allPublic”, “allPrivate”
- **bucket_info** (*dict*) – additional bucket info to store with the bucket
- **cors_rules** (*dict*) – bucket CORS rules to store with the bucket
- **lifecycle_rules** (*dict*) – bucket lifecycle rules to store with the bucket

Returns a Bucket object

Return type *b2sdk.bucket.Bucket*

create_key (*capabilities, key_name, valid_duration_seconds=None, bucket_id=None, name_prefix=None*)

Create a new application key

Parameters

- **capabilities** (*list*) – a list of capabilities
- **key_name** (*str*) – a name of a key
- **valid_duration_seconds** (*int*) – key duration in seconds
- **bucket_id** (*str*) – a bucket ID
- **name_prefix** (*str*) – a name prefix

delete_bucket (*bucket*)

Deletes the bucket remotely. For legacy reasons it returns whatever server sends in response, but API user should not rely on the response: if it doesn’t raise an exception, it means that the operation was a success

Parameters `bucket` (`b2sdk.bucket.Bucket`) – a Bucket object

delete_file_version (`file_id`, `file_name`)

Permanently and irrevocably delete one version of a file

Parameters

- **file_id** – a file ID
- **file_name** (`str`) – a file name

Type `file_id`: `str`

delete_key (`application_key_id`)

Delete application key with a given ID

Parameters `application_key_id` – an application key ID

download_file_by_id (`file_id`, `download_dest`, `progress_listener=None`, `range_=None`)

Download a file with a given ID

Parameters

- **file_id** (`str`) – a file ID
- **download_dest** (`str`) – a local file path
- **progress_listener** – an instance of the one of the following classes: `PartProgressReporter`, `TqdmProgressListener`, `SimpleProgressListener`, `DoNothingProgressListener`, `ProgressListenerForTest`, `SyncFileReporter`, or any sub class of `AbstractProgressListener`
- **range** (`list`) – a list of two integers, the first one is a start position, and the second one is the end position in the file

Returns context manager that returns an object that supports `iter_content()`

get_account_id ()

Return account ID

Returns account ID

Return type `str`

get_bucket_by_id (`bucket_id`)

Return bucket object with a given ID

Parameters `bucket_id` (`str`) – a bucket ID

Returns a Bucket object

Return type `b2sdk.bucket.Bucket`

get_bucket_by_name (`bucket_name`)

Returns the Bucket for the given bucket_name.

Parameters `bucket_name` (`str`) – The name of the bucket to return.

Returns a Bucket object

Return type `b2sdk.bucket.Bucket`

Raises `b2sdk.exception.NonExistentBucket` – if the bucket does not exist in the account

get_download_url_for_file_name (*bucket_name*, *file_name*)

Returns a URL to download the given file by name.

Parameters

- **bucket_name** (*str*) – a bucket name
- **file_name** (*str*) – a file name

get_download_url_for_fileid (*file_id*)

Returns a URL to download the given file by ID.

Parameters **file_id** (*str*) – a file ID

get_file_info (*file_id*)

legacy interface which just returns whatever remote API returns

get_thread_pool ()

Returns the thread pool executor to use for uploads and downloads.

list_buckets (*bucket_name=None*)

Calls `b2_list_buckets` and returns a list of buckets.

When no bucket name is specified, returns *all* of the buckets in the account. When a bucket name is given, returns just that bucket. When authorized with an application key restricted to one bucket, you must specify the bucket name, or the request will be unauthorized.

Parameters **bucket_name** (*str*) – the name of the one bucket to return.

Returns A list of instances of `b2sdk.bucket.Bucket`.

Return type `list`

list_keys (*start_application_key_id=None*)

List application keys

Parameters **start_application_key_id** (*str*) – an application key ID to start from

list_parts (*file_id*, *start_part_number=None*, *batch_size=None*)

Generator that yields a `Part` for each of the parts that have been uploaded.

Parameters

- **file_id** (*str*) – the ID of the large file that is not finished
- **start_part_number** (*int*) – the first part number to return. defaults to the first part.
- **batch_size** (*int*) – the number of parts to fetch at a time from the server

set_thread_pool_size (*max_workers*)

Sets the size of the thread pool to use for uploads and downloads.

Must be called before any work starts, or the thread pool will get the default size of 1.

Parameters **max_workers** (*int*) – maximum allowed number of workers in a pool

`b2sdk.api.url_for_api` (*info*, *api_name*)

Return URL for an API endpoint

Parameters

- **info** – account info
- **api_name** (*str*) –

Return type `str`

4.1.4 b2sdk.bucket – Bucket

```
class b2sdk.bucket.Bucket (api, id_, name=None, type_=None, bucket_info=None,  
                           cors_rules=None, lifecycle_rules=None, revision=None,  
                           bucket_dict=None)
```

Bases: `object`

Provides access to a bucket in B2: listing files, uploading and downloading.

```
DEFAULT_CONTENT_TYPE = 'b2/x-auto'
```

```
MAX_LARGE_FILE_SIZE = 10000000000000
```

```
MAX_UPLOAD_ATTEMPTS = 5
```

```
__init__ (api, id_, name=None, type_=None, bucket_info=None, cors_rules=None, lifecy-  
          cle_rules=None, revision=None, bucket_dict=None)
```

Parameters

- **api** (`b2sdk.api.B2Api`) – an API object
- **id** (`str`) – a bucket id
- **name** (`str`) – a bucket name
- **type** (`str`) – a bucket type
- **bucket_info** (`dict`) – an info to store with a bucket
- **cors_rules** (`dict`) – CORS rules to store with a bucket
- **lifecycle_rules** (`dict`) – lifecycle rules to store with a bucket
- **revision** (`int`) – a bucket revision number
- **bucket_dict** (`dict`) – a dictionary which contains bucket parameters

```
as_dict ()
```

Return bucket representation as a dictionary

Return type `dict`

```
cancel_large_file (file_id)
```

Cancel large file transfer

Parameters **file_id** (`str`) – a file ID

```
copy_file (file_id, new_file_name, bytes_range=None, metadata_directive=None, con-  
          tent_type=None, file_info=None, destination_bucket_id=None)
```

Creates a new file in this bucket by (server-side) copying from an existing file.

Parameters

- **file_id** (`str`) – file ID of existing file
- **new_file_name** (`str`) – file name of the new file
- **bytes_range** (`tuple[int, int]`, `None`) – start and end offsets, default is the entire file
- **metadata_directive** (`b2sdk.v1.MetadataDirectiveMode`, `None`) – default is `b2sdk.v1.MetadataDirectiveMode.COPY`
- **content_type** (`str`, `None`) – `content_type` for the new file if `metadata_directive` is set to `b2sdk.v1.MetadataDirectiveMode.REPLACE`, default will copy the `content_type` of old file

- **file_info** (*dict*, *None*) – file_info for the new file if metadata_directive is set to `b2sdk.v1.MetadataDirectiveMode.REPLACE`, default will copy the file_info of old file

delete_file_version (*file_id*, *file_name*)

Delete file version

Parameters

- **file_id** (*str*) – a file ID
- **file_name** (*str*) – a file name

download_file_by_id (*file_id*, *download_dest*, *progress_listener=None*, *range_=None*)

Download a file by ID, `download_file_by_id` actually belongs in B2Api, not in Bucket, we just provide a convenient redirect here

Parameters

- **file_id** (*str*) – a file ID
- **download_dest** (*str*) – a local file path
- **progress_listener** (`b2sdk.progress.AbstractProgressListener`) – a progress listener object
- **range** (*list*) – a list of two integer values, start and end offsets

download_file_by_name (*file_name*, *download_dest*, *progress_listener=None*, *range_=None*)

Download a file by name

Parameters

- **file_id** (*str*) – a file ID
- **download_dest** (*str*) – a local file path
- **progress_listener** (`b2sdk.progress.AbstractProgressListener`) – a progress listener object
- **range** (*list*) – a list of two integer values, start and end offsets

get_download_authorization (*file_name_prefix*, *valid_duration_in_seconds*)

Return an authorization token that is valid only for downloading files from the given bucket

Parameters

- **file_name_prefix** (*str*) – a file name prefix, only files that match it could be downloaded
- **valid_duration_in_seconds** (*int*) – a token is valid only during this amount of seconds

get_download_url (*filename*)

Get file download URL

Parameters **filename** (*str*) – a file name

Returns file download URL

Return type *str*

get_id ()

Return bucket ID

Type *str*

hide_file (*file_name*)

Hide a file

Parameters **file_name** (*str*) – a file name

Returns file version info

Return type *b2sdk.file_version.FileVersionInfoFactory*

list_file_names (*start_filename=None, max_entries=None, prefix=None*)

legacy interface which just returns whatever remote API returns

list_file_versions (*start_filename=None, start_file_id=None, max_entries=None, prefix=None*)

legacy interface which just returns whatever remote API returns

list_parts (*file_id, start_part_number=None, batch_size=None*)

Get a list of all parts that have been uploaded for a given file

Parameters

- **file_id** (*str*) – a file ID
- **start_part_number** (*int*) – the first part number to return. defaults to the first part.
- **batch_size** (*int*) – the number of parts to fetch at a time from the server

list_unfinished_large_files (*start_file_id=None, batch_size=None*)

A generator that yields an UnfinishedLargeFile for each unfinished large file in the bucket, starting at the given file.

Parameters

- **start_file_id** (*str*) – a file ID to start with
- **batch_size** (*int*) – max file count

Return type generator

ls (*folder_to_list="", show_versions=False, recursive=False, fetch_count=None*)

Pretends that folders exist, and yields the information about the files in a folder.

B2 has a flat namespace for the files in a bucket, but there is a convention of using “/” as if there were folders. This method searches through the flat namespace to find the files and “folders” that live within a given folder.

When the *recursive* flag is set, lists all of the files in the given folder, and all of its sub-folders.

Parameters

- **folder_to_list** (*str*) – the name of the folder to list. Must not start with “/”. Empty string means top-level folder.
- **show_versions** (*bool*) – when true returns info about all versions of a file, when false, just returns info about the most recent versions.
- **recursive** (*bool*) – if True, list folders recursively
- **fetch_count** (*int*) – how many entries to return. 1 - 1000

Return type generator

set_info (*new_bucket_info, if_revision_is=None*)

Update bucket info

Parameters

- **new_bucket_info** (*dict*) – new bucket info dictionary

- **if_revision_is** (*int*) – revision number, update the info if revision equals to *if_revision_is*' value

set_type (*bucket_type*)

Update bucket type

Parameters **bucket_type** (*str*) – a bucket type

start_large_file (*file_name*, *content_type=None*, *file_info=None*)

Start large file transfer

Parameters

- **file_name** (*str*) – a file name
- **content_type** (*str*) – request content type
- **file_info** (*dict*) – a file info to store with the file

update (*bucket_type=None*, *bucket_info=None*, *cors_rules=None*, *lifecycle_rules=None*, *if_revision_is=None*)

Update various bucket parameters

Parameters

- **bucket_type** (*str*) – a bucket type
- **bucket_info** (*dict*) – an info to store with a bucket
- **cors_rules** (*dict*) – CORS rules to store with a bucket
- **lifecycle_rules** (*dict*) – lifecycle rules to store with a bucket
- **if_revision_is** (*int*) – revision number, update the info if revision equals to *if_revision_is*' value

upload (*upload_source*, *file_name*, *content_type=None*, *file_info=None*, *min_part_size=None*, *progress_listener=None*)

Uploads a file to B2, retrying as needed.

The source of the upload is an UploadSource object that can be used to open (and re-open) the file. The result of opening should be a binary file whose read() method returns bytes.

Parameters

- **upload_source** (*b2sdk.upload_source.UploadSource*) – an UploadSource object that opens the source of the upload
- **file_name** (*str*) – the file name of the new B2 file
- **content_type** (*str*, *None*) – the MIME type, or None to accept the default based on file extension of the B2 file name
- **file_infos** (*dict*) – custom file info to be stored with the file
- **min_part_size** (*int*) – the smallest part size to use
- **progress_listener** (*b2sdk.progress.AbstractProgressListener*) – object to notify as data is transferred

The function *opener* should return a file-like object, and it must be possible to call it more than once in case the upload is retried.

upload_bytes (*data_bytes*, *file_name*, *content_type=None*, *file_infos=None*, *progress_listener=None*)

Upload bytes in memory to a B2 file

Parameters

- **data_bytes** (*bytes*) – a byte array to upload
- **file_name** (*str*) – a file name to upload bytes to
- **content_type** (*str*) – request content type
- **file_infos** (*dict*) – a file info to store with the file
- **progress_listener** (`b2sdk.progress.AbstractProgressListener`) – a progress listener object

upload_local_file (*local_file, file_name, content_type=None, file_infos=None, sha1_sum=None, min_part_size=None, progress_listener=None*)

Uploads a file on local disk to a B2 file.

Parameters

- **local_file** (*str*) – a path to a file on local disk
- **file_name** (*str*) – a file name of the new B2 file
- **content_type** (*str*) – request content type
- **file_infos** (*dict*) – a file info to store with the file
- **sha1_sum** (*str*) – file SHA1 hash
- **min_part_size** (*int*) – a minimum size of a part
- **progress_listener** (`b2sdk.progress.AbstractProgressListener`) – a progress listener object

class `b2sdk.bucket.BucketFactory`

Bases: `object`

This is a factory for creating bucket objects from another kind of objects

classmethod `from_api_bucket_dict` (*api, bucket_dict*)

Turns a dictionary, like this:

```
{
    "bucketType": "allPrivate",
    "bucketId": "a4ba6a39d8b6b5fd561f0010",
    "bucketName": "zsdfrtsazsdfafr",
    "accountId": "4aa9865d6f00",
    "bucketInfo": {},
    "revision": 1
}
```

into a `Bucket` object

Parameters

- **api** (`b2sdk.api.B2Api`) – API object
- **bucket_dict** (*dict*) – a dictionary with bucket properties

Returns a `Bucket` object

Return type `b2sdk.bucket.Bucket`

classmethod `from_api_response` (*api, response*)

Create a `Bucket` object from API response

Parameters

- **api** (`b2sdk.api.B2Api`) – API object
- **response** (*object*) – response object

Returns a Bucket object

Return type `b2sdk.bucket.Bucket`

class `b2sdk.bucket.LargeFileUploadState` (*file_progress_listener*)

Bases: `object`

Tracks the status of uploading a large file, accepting updates from the tasks that upload each of the parts.

The aggregated progress is passed on to a ProgressListener that reports the progress for the file as a whole.

This class is THREAD SAFE.

__init__ (*file_progress_listener*)

Parameters **file_progress_listener** (`b2sdk.progress.AbstractProgressListener`) – a progress listener object

get_error_message ()

Get an error message

Returns an error message

Return type `str`

has_error ()

Check whether an error occurred

Return type `bool`

set_error (*message*)

Set error message

Parameters **message** (`str`) – an error message

update_part_bytes (*bytes_delta*)

Update listener progress info

Parameters **bytes_delta** (`int`) – number of bytes to increase a progress for

class `b2sdk.bucket.PartProgressReporter` (*large_file_upload_state, *args, **kwargs*)

Bases: `b2sdk.progress.AbstractProgressListener`

An adapter that listens to the progress of upload a part and gives the information to a LargeFileUploadState.

Accepts absolute bytes_completed from the uploader, and reports deltas to the LargeFileUploadState. The bytes_completed for the part will drop back to 0 on a retry, which will result in a negative delta.

__init__ (*large_file_upload_state, *args, **kwargs*)

Parameters **large_file_upload_state** (`b2sdk.bucket.LargeFileUploadState`) – LargeFileUploadState object

bytes_completed (*byte_count*)

Update bytes completed value

Parameters **bytes_count** (`int`) – a value to increase progress to

close ()

Perform cleanup operations

set_total_bytes (*total_byte_count*)

Set total bytes value

Parameters `total_bytes_count` (*int*) –

4.1.5 b2sdk.cache – Cache

```
class b2sdk.cache.AbstractCache
    Bases: object

    clear()

    abstract get_bucket_id_or_none_from_bucket_name(name)

    abstract get_bucket_name_or_none_from_allowed()

    abstract save_bucket(bucket)

    abstract set_bucket_name_cache(buckets)

class b2sdk.cache.AuthInfoCache(info)
    Bases: b2sdk.cache.AbstractCache

    Cache that stores data persistently in StoredAccountInfo

    __init__(info)
        Initialize self. See help(type(self)) for accurate signature.

    get_bucket_id_or_none_from_bucket_name(name)

    get_bucket_name_or_none_from_allowed()

    save_bucket(bucket)

    set_bucket_name_cache(buckets)

class b2sdk.cache.DummyCache
    Bases: b2sdk.cache.AbstractCache

    Cache that does nothing

    get_bucket_id_or_none_from_bucket_name(name)

    get_bucket_name_or_none_from_allowed()

    save_bucket(bucket)

    set_bucket_name_cache(buckets)

class b2sdk.cache.InMemoryCache
    Bases: b2sdk.cache.AbstractCache

    Cache that stores the information in memory

    __init__()
        Initialize self. See help(type(self)) for accurate signature.

    get_bucket_id_or_none_from_bucket_name(name)

    get_bucket_name_or_none_from_allowed()

    save_bucket(bucket)

    set_bucket_name_cache(buckets)
```

4.1.6 b2sdk.progress – Progress

class b2sdk.progress.**AbstractProgressListener**

Bases: `object`

Interface expected by B2Api upload and download methods to report on progress.

This interface just accepts the number of bytes transferred so far. Subclasses will need to know the total size if they want to report a percent done.

__init__ ()

Initialize self. See help(type(self)) for accurate signature.

abstract bytes_completed (*byte_count*)

Reports that the given number of bytes have been transferred so far. This is not a delta, it is the total number of bytes transferred so far.

Parameters *byte_count* (*int*) – number of bytes have been transferred

abstract close ()

Must be called when you're done with the listener. In well-structured code, should be called only once.

abstract set_total_bytes (*total_byte_count*)

Always called before `__enter__` to set the expected total number of bytes.

May be called more than once if an upload is retried.

Parameters *total_byte_count* (*int*) – expected total number of bytes

class b2sdk.progress.**AbstractStreamWithProgress** (*stream*, *progress_listener*, *offset=0*)

Bases: `object`

Wraps a file-like object and updates a ProgressListener as data is read / written. In the abstract class, read and write methods do not update the progress - child classes shall do it

__init__ (*stream*, *progress_listener*, *offset=0*)

Parameters

- **stream** – the stream to read from or write to
- **progress_listener** (`b2sdk.progress.AbstractProgressListener`) – the listener that we tell about progress
- **offset** (*int*) – the starting byte offset in the file

flush ()

Flush the stream

read (*size=None*)

Read data from the stream

Parameters *size* (*int*) – number of bytes to read

Returns data read from the stream

seek (*pos*)

Seek to a given position in the stream

Parameters *pos* (*int*) – position in the stream

tell ()

Return current stream position

Return type *int*

write (*data*)
Write data to the stream

Parameters **data** – a data to write to the stream

class `b2sdk.progress.DoNothingProgressListener`
Bases: `b2sdk.progress.AbstractProgressListener`

This listener performs no any output

bytes_completed (*byte_count*)
Reports that the given number of bytes have been transferred so far. This is not a delta, it is the total number of bytes transferred so far.

Parameters **byte_count** (*int*) – number of bytes have been transferred

close ()
Perform clean up operations

set_total_bytes (*total_byte_count*)
Set the expected total number of bytes.

Parameters **total_byte_count** (*int*) – expected total number of bytes

class `b2sdk.progress.ProgressListenerForTest` (**args, **kwargs*)
Bases: `b2sdk.progress.AbstractProgressListener`

Captures all of the calls so they can be checked.

__init__ (**args, **kwargs*)
Initialize self. See help(type(self)) for accurate signature.

bytes_completed (*byte_count*)
Reports that the given number of bytes have been transferred so far. This is not a delta, it is the total number of bytes transferred so far.

Parameters **byte_count** (*int*) – number of bytes have been transferred

close ()
Must be called when you're done with the listener. In well-structured code, should be called only once.

get_calls ()

set_total_bytes (*total_byte_count*)
Always called before **__enter__** to set the expected total number of bytes.
May be called more than once if an upload is retried.

Parameters **total_byte_count** (*int*) – expected total number of bytes

class `b2sdk.progress.RangeOfInputStream` (*stream, offset, length*)
Bases: `object`

Wraps a file-like object (read only) and reads the selected range of the file.

__init__ (*stream, offset, length*)

Parameters

- **stream** – a seekable stream
- **offset** (*int*) – offset in the stream
- **length** (*int*) – max number of bytes to read

read (*size=None*)
Read data from the stream

Parameters **size** (*int*) – number of bytes to read

Returns data read from the stream

seek (*pos*)

Seek to a given position in the stream

Parameters **pos** (*int*) – position in the stream

class `b2sdk.progress.ReadingStreamWithProgress` (*stream, progress_listener, offset=0*)

Bases: `b2sdk.progress.AbstractStreamWithProgress`

Wraps a file-like object, updates progress while reading

read (*size=None*)

Read data from the stream

Parameters **size** (*int*) – number of bytes to read

Returns data read from the stream

class `b2sdk.progress.SimpleProgressListener` (*description, *args, **kwargs*)

Bases: `b2sdk.progress.AbstractProgressListener`

Just a simple progress listener which prints info on a console

__init__ (*description, *args, **kwargs*)

Initialize self. See help(type(self)) for accurate signature.

bytes_completed (*byte_count*)

Reports that the given number of bytes have been transferred so far. This is not a delta, it is the total number of bytes transferred so far.

Parameters **byte_count** (*int*) – number of bytes have been transferred

close ()

Perform clean up operations

set_total_bytes (*total_byte_count*)

Set the expected total number of bytes.

Parameters **total_byte_count** (*int*) – expected total number of bytes

class `b2sdk.progress.StreamWithHash` (*stream*)

Bases: `object`

Wraps a file-like object, calculates SHA1 while reading and appends hash at the end

__init__ (*stream*)

Parameters **stream** – the stream to read from

Returns None

hash_size ()

Calculate size of a hash string

Return type `int`

read (*size=None*)

Read data from the stream

Parameters **size** (*int*) – number of bytes to read

Returns data read from the stream

seek (*pos*)

Seek to a given position in the stream

Parameters *pos* (*int*) – position in the stream

class `b2sdk.progress.TqdmProgressListener` (*description*, **args*, ***kwargs*)

Bases: `b2sdk.progress.AbstractProgressListener`

Progress listener based on tqdm library

__init__ (*description*, **args*, ***kwargs*)

Initialize self. See help(type(self)) for accurate signature.

bytes_completed (*byte_count*)

Reports that the given number of bytes have been transferred so far. This is not a delta, it is the total number of bytes transferred so far.

Parameters *byte_count* (*int*) – number of bytes have been transferred

close ()

Perform clean up operations

set_total_bytes (*total_byte_count*)

Set the expected total number of bytes.

Parameters *total_byte_count* (*int*) – expected total number of bytes

class `b2sdk.progress.WritingStreamWithProgress` (*stream*, *progress_listener*, *offset=0*)

Bases: `b2sdk.progress.AbstractStreamWithProgress`

Wraps a file-like object, updates progress while writing

write (*data*)

Write data to the stream

Parameters *data* – a data to write to the stream

`b2sdk.progress.make_progress_listener` (*description*, *quiet*)

Returns a progress listener object depending on some conditions

Parameters

- **description** (*str*) – listener description
- **quiet** (*bool*) – if True, do not output anything

Returns a listener object

4.1.7 b2sdk.exception – Exceptions

exception `b2sdk.exception.AlreadyFailed` (**args*, ***kwargs*)

Bases: `b2sdk.exception.B2SimpleError`

exception `b2sdk.exception.B2ConnectionError` (**args*, ***kwargs*)

Bases: `b2sdk.exception.TransientErrorMixin`, `b2sdk.exception.B2SimpleError`

exception `b2sdk.exception.B2Error` (**args*, ***kwargs*)

Bases: `Exception`

__init__ (**args*, ***kwargs*)

Python 2 does not like it when you pass unicode as the message in an exception. We like to use file names in exception messages. To avoid problems, if the message has any non-ascii characters in it, they are replaced with backslash-uNNNN

<https://pythonhosted.org/kitchen/unicode-frustrations.html#frustration-5-exceptions>

property prefix

nice auto-generated error message prefix >>> B2SimpleError().prefix 'Simple error' >>> AlreadyFailed().prefix 'Already failed'

should_retry_http()

Returns true if this is an error that can cause an HTTP call to be retried.

should_retry_upload()

Returns true if this is an error that should tell the upload code to get a new upload URL and try the upload again.

exception b2sdk.exception.B2HttpCallbackException(*args, **kwargs)

Bases: *b2sdk.exception.B2SimpleError*

exception b2sdk.exception.B2HttpCallbackPostRequestException(*args, **kwargs)

Bases: *b2sdk.exception.B2HttpCallbackException*

exception b2sdk.exception.B2HttpCallbackPreRequestException(*args, **kwargs)

Bases: *b2sdk.exception.B2HttpCallbackException*

exception b2sdk.exception.B2RequestTimeout(*args, **kwargs)

Bases: *b2sdk.exception.TransientErrorMixin, b2sdk.exception.B2SimpleError*

exception b2sdk.exception.B2SimpleError(*args, **kwargs)

Bases: *b2sdk.exception.B2Error*

a B2Error with a message prefix

exception b2sdk.exception.BadDateFormat(*args, **kwargs)

Bases: *b2sdk.exception.B2SimpleError*

prefix = 'Date from server'

exception b2sdk.exception.BadFileInfo(*args, **kwargs)

Bases: *b2sdk.exception.B2SimpleError*

exception b2sdk.exception.BadJson(*args, **kwargs)

Bases: *b2sdk.exception.B2SimpleError*

prefix = 'Bad request'

exception b2sdk.exception.BadUploadUrl(*args, **kwargs)

Bases: *b2sdk.exception.TransientErrorMixin, b2sdk.exception.B2SimpleError*

exception b2sdk.exception.BrokenPipe(*args, **kwargs)

Bases: *b2sdk.exception.B2Error*

should_retry_upload()

Returns true if this is an error that should tell the upload code to get a new upload URL and try the upload again.

exception b2sdk.exception.BucketNotAllowed(*args, **kwargs)

Bases: *b2sdk.exception.NotAllowedByAppKeyError*

exception b2sdk.exception.CapabilityNotAllowed(*args, **kwargs)

Bases: *b2sdk.exception.NotAllowedByAppKeyError*

exception b2sdk.exception.ChecksumMismatch(checksum_type, expected, actual)

Bases: *b2sdk.exception.TransientErrorMixin, b2sdk.exception.B2Error*

__init__(checksum_type, expected, actual)

Initialize self. See help(type(self)) for accurate signature.

exception `b2sdk.exception.ClockSkew(clock_skew_seconds)`
 Bases: `b2sdk.exception.B2HttpCallbackPostRequestException`

The clock on the server differs from the local clock by too much.

`__init__` (*clock_skew_seconds*)

Parameters **clock_skew_seconds** – The different: local_clock - server_clock

exception `b2sdk.exception.CommandError(message)`
 Bases: `b2sdk.exception.B2Error`

b2 command error (user caused). Accepts exactly one argument. We expect users of shell scripts will parse our `__str__` output.

`__init__` (*message*)

Python 2 does not like it when you pass unicode as the message in an exception. We like to use file names in exception messages. To avoid problems, if the message has any non-ascii characters in it, they are replaced with backslash-uNNNN

<https://pythonhosted.org/kitchen/unicode-frustrations.html#frustration-5-exceptions>

exception `b2sdk.exception.Conflict(*args, **kwargs)`
 Bases: `b2sdk.exception.B2SimpleError`

exception `b2sdk.exception.ConnectionReset(*args, **kwargs)`
 Bases: `b2sdk.exception.B2Error`

`should_retry_upload()`

Returns true if this is an error that should tell the upload code to get a new upload URL and try the upload again.

exception `b2sdk.exception.DestFileNewer(dest_file, source_file, dest_prefix, source_prefix)`
 Bases: `b2sdk.exception.B2Error`

`__init__` (*dest_file*, *source_file*, *dest_prefix*, *source_prefix*)

Python 2 does not like it when you pass unicode as the message in an exception. We like to use file names in exception messages. To avoid problems, if the message has any non-ascii characters in it, they are replaced with backslash-uNNNN

<https://pythonhosted.org/kitchen/unicode-frustrations.html#frustration-5-exceptions>

`should_retry_http()`

Returns true if this is an error that can cause an HTTP call to be retried.

exception `b2sdk.exception.DuplicateBucketName(*args, **kwargs)`
 Bases: `b2sdk.exception.B2SimpleError`

`prefix` = 'Bucket name is already in use'

exception `b2sdk.exception.FileAlreadyHidden(*args, **kwargs)`
 Bases: `b2sdk.exception.B2SimpleError`

exception `b2sdk.exception.FileNameNotAllowed(*args, **kwargs)`
 Bases: `b2sdk.exception.NotAllowedByAppKeyError`

exception `b2sdk.exception.FileNotPresent(*args, **kwargs)`
 Bases: `b2sdk.exception.B2SimpleError`

exception `b2sdk.exception.InvalidAuthToken(message, code)`
 Bases: `b2sdk.exception.Unauthorized`

Specific type of Unauthorized that means the auth token is invalid. This is not the case where the auth token is valid but does not allow access.

__init__ (*message, code*)

Python 2 does not like it when you pass unicode as the message in an exception. We like to use file names in exception messages. To avoid problems, if the message has any non-ascii characters in it, they are replaced with backslash-uNNNN

<https://pythonhosted.org/kitchen/unicode-frustrations.html#frustration-5-exceptions>

exception `b2sdk.exception.InvalidMetadataDirective` (**args, **kwargs*)

Bases: `b2sdk.exception.B2Error`

exception `b2sdk.exception.InvalidRange` (*content_length, range_*)

Bases: `b2sdk.exception.B2Error`

__init__ (*content_length, range_*)

Python 2 does not like it when you pass unicode as the message in an exception. We like to use file names in exception messages. To avoid problems, if the message has any non-ascii characters in it, they are replaced with backslash-uNNNN

<https://pythonhosted.org/kitchen/unicode-frustrations.html#frustration-5-exceptions>

exception `b2sdk.exception.InvalidUploadSource` (**args, **kwargs*)

Bases: `b2sdk.exception.B2SimpleError`

exception `b2sdk.exception.MaxFileSizeExceeded` (*size, max_allowed_size*)

Bases: `b2sdk.exception.B2Error`

__init__ (*size, max_allowed_size*)

Python 2 does not like it when you pass unicode as the message in an exception. We like to use file names in exception messages. To avoid problems, if the message has any non-ascii characters in it, they are replaced with backslash-uNNNN

<https://pythonhosted.org/kitchen/unicode-frustrations.html#frustration-5-exceptions>

exception `b2sdk.exception.MaxRetriesExceeded` (*limit, exception_info_list*)

Bases: `b2sdk.exception.B2Error`

__init__ (*limit, exception_info_list*)

Python 2 does not like it when you pass unicode as the message in an exception. We like to use file names in exception messages. To avoid problems, if the message has any non-ascii characters in it, they are replaced with backslash-uNNNN

<https://pythonhosted.org/kitchen/unicode-frustrations.html#frustration-5-exceptions>

exception `b2sdk.exception.MissingPart` (**args, **kwargs*)

Bases: `b2sdk.exception.B2SimpleError`

prefix = 'Part number has not been uploaded'

exception `b2sdk.exception.NonExistentBucket` (**args, **kwargs*)

Bases: `b2sdk.exception.B2SimpleError`

prefix = 'No such bucket'

exception `b2sdk.exception.NotAllowedByAppKeyError` (**args, **kwargs*)

Bases: `b2sdk.exception.B2SimpleError`

Base class for errors caused by restrictions on an application key.

exception `b2sdk.exception.PartShalMismatch` (*key*)

Bases: `b2sdk.exception.B2Error`

__init__ (*key*)

Python 2 does not like it when you pass unicode as the message in an exception. We like to use file names

in exception messages. To avoid problems, if the message has any non-ascii characters in it, they are replaced with backslash-uNNNN

<https://pythonhosted.org/kitchen/unicode-frustrations.html#frustration-5-exceptions>

exception `b2sdk.exception.RestrictedBucket (bucket_name)`

Bases: `b2sdk.exception.B2Error`

`__init__ (bucket_name)`

Python 2 does not like it when you pass unicode as the message in an exception. We like to use file names in exception messages. To avoid problems, if the message has any non-ascii characters in it, they are replaced with backslash-uNNNN

<https://pythonhosted.org/kitchen/unicode-frustrations.html#frustration-5-exceptions>

exception `b2sdk.exception.ServiceError (*args, **kwargs)`

Bases: `b2sdk.exception.TransientErrorMixin`, `b2sdk.exception.B2Error`

Used for HTTP status codes 500 through 599.

exception `b2sdk.exception.StorageCapExceeded (*args, **kwargs)`

Bases: `b2sdk.exception.B2Error`

exception `b2sdk.exception.TooManyRequests (*args, **kwargs)`

Bases: `b2sdk.exception.B2Error`

`should_retry_http ()`

Returns true if this is an error that can cause an HTTP call to be retried.

class `b2sdk.exception.TransientErrorMixin`

Bases: `object`

`should_retry_http ()`

`should_retry_upload ()`

exception `b2sdk.exception.TruncatedOutput (bytes_read, file_size)`

Bases: `b2sdk.exception.TransientErrorMixin`, `b2sdk.exception.B2Error`

`__init__ (bytes_read, file_size)`

Initialize self. See help(type(self)) for accurate signature.

exception `b2sdk.exception.Unauthorized (message, code)`

Bases: `b2sdk.exception.B2Error`

`__init__ (message, code)`

Python 2 does not like it when you pass unicode as the message in an exception. We like to use file names in exception messages. To avoid problems, if the message has any non-ascii characters in it, they are replaced with backslash-uNNNN

<https://pythonhosted.org/kitchen/unicode-frustrations.html#frustration-5-exceptions>

`should_retry_upload ()`

Returns true if this is an error that should tell the upload code to get a new upload URL and try the upload again.

exception `b2sdk.exception.UnexpectedCloudBehaviour (*args, **kwargs)`

Bases: `b2sdk.exception.B2SimpleError`

exception `b2sdk.exception.UnknownError (*args, **kwargs)`

Bases: `b2sdk.exception.B2SimpleError`

exception `b2sdk.exception.UnknownHost (*args, **kwargs)`

Bases: `b2sdk.exception.B2Error`

```
exception b2sdk.exception.UnrecognizedBucketType (*args, **kwargs)
    Bases: b2sdk.exception.B2Error
```

```
exception b2sdk.exception.UnsatisfiableRange (*args, **kwargs)
    Bases: b2sdk.exception.B2Error
```

```
exception b2sdk.exception.UnusableFileName (*args, **kwargs)
    Bases: b2sdk.exception.B2SimpleError
```

Raise when a filename doesn't meet the rules.

Could possibly use InvalidUploadSource, but this is intended for the filename on the server, which could differ.
<https://www.backblaze.com/b2/docs/files.html>.

```
b2sdk.exception.interpret_b2_error (status, code, message, post_params=None)
```

4.1.8 b2sdk.file_version – File version

```
class b2sdk.file_version.FileIdAndName (file_id, file_name)
    Bases: object
```

```
__init__ (file_id, file_name)
    Initialize self. See help(type(self)) for accurate signature.
```

```
as_dict ()
```

```
class b2sdk.file_version.FileVersionInfo (id_, file_name, size, content_type, content_sha1,
                                           file_info, upload_timestamp, action)
```

Bases: *object*

```
LS_ENTRY_TEMPLATE = '%83s %6s %10s %8s %9d %s'
```

```
__init__ (id_, file_name, size, content_type, content_sha1, file_info, upload_timestamp, action)
    Initialize self. See help(type(self)) for accurate signature.
```

```
as_dict ()
```

```
classmethod format_folder_ls_entry (name)
```

```
format_ls_entry ()
```

```
class b2sdk.file_version.FileVersionInfoFactory
    Bases: object
```

```
classmethod from_api_response (file_info_dict, force_action=None)
    turns this:
```

```
{
    "action": "hide",
    "fileId": "4_zBucketName_f103b7ca31313c69c_d20151230_m030117_c001_
↪v0001015_t0000",
    "fileName": "randomdata",
    "size": 0,
    "uploadTimestamp": 1451444477000
}
```

or this:

```
{
    "accountId": "4aa9865d6f00",
    "bucketId": "547a2a395826655d561f0010",

```

(continues on next page)

(continued from previous page)

```

    "contentLength": 1350,
    "contentShal": "753calc2d0f3e8748320b38f5da057767029a036",
    "contentType": "application/octet-stream",
    "fileId": "4_z547a2a395826655d561f0010_f106d4ca95f8b5b78_d20160104_
↪m003906_c001_v0001013_t0005",
    "fileInfo": {},
    "fileName": "randomdata"
}

```

into a FileVersionInfo object

classmethod `from_cancel_large_file_response` (*response*)

4.1.9 Sync

b2sdk.sync.action – Sync action

class `b2sdk.sync.action.AbstractAction`

Bases: `object`

An action to take, such as uploading, downloading, or deleting a file. Multi-threaded tasks create a sequence of Actions, which are then run by a pool of threads.

An action can depend on other actions completing. An example of this is making sure a CreateBucketAction happens before an UploadFileAction.

abstract `do_action` (*bucket*, *reporter*)

Performs the action, returning only after the action is completed.

Parameters

- **bucket** (`b2sdk.bucket.Bucket`) – a Bucket object
- **reporter** – a place to report errors

abstract `do_report` (*bucket*, *reporter*)

Report the action performed.

Parameters

- **bucket** (`b2sdk.bucket.Bucket`) – a Bucket object
- **reporter** – a place to report errors

abstract `get_bytes` ()

Returns the number of bytes to transfer for this action.

Return type

`int`

run (*bucket*, *reporter*, *dry_run=False*)

Main action routine

Parameters

- **bucket** (`b2sdk.bucket.Bucket`) – a Bucket object
- **reporter** – a place to report errors
- **dry_run** (*bool*) – if True, perform a dry run

class `b2sdk.sync.action.B2DeleteAction` (*relative_name*, *b2_file_name*, *file_id*, *note*)

Bases: `b2sdk.sync.action.AbstractAction`

`__init__(relative_name, b2_file_name, file_id, note)`

Parameters

- **relative_name** (*str*) – a relative file name
- **b2_file_name** (*str*) – a name of a remote file
- **file_id** (*str*) – a file ID
- **note** (*str*) – a deletion note

`do_action(bucket, reporter)`

Performs the deleting action, returning only after the action is completed.

Parameters

- **bucket** (`b2sdk.bucket.Bucket`) – a Bucket object
- **reporter** – a place to report errors

`do_report(bucket, reporter)`

Report the deleting action performed.

Parameters

- **bucket** (`b2sdk.bucket.Bucket`) – a Bucket object
- **reporter** – a place to report errors

`get_bytes()`

Return file size

Returns always zero

Return type `int`

`class b2sdk.sync.action.B2DownloadAction(relative_name, b2_file_name, file_id, local_full_path, mod_time_millis, file_size)`

Bases: `b2sdk.sync.action.AbstractAction`

`__init__(relative_name, b2_file_name, file_id, local_full_path, mod_time_millis, file_size)`

Parameters

- **relative_name** (*str*) – a relative file name
- **b2_file_name** (*str*) – a name of a remote file
- **file_id** (*str*) – a file ID
- **local_full_path** (*str*) – a local file path
- **mod_time_millis** (*int*) – file modification time in milliseconds
- **file_size** (*int*) – a file size

`do_action(bucket, reporter)`

Performs the downloading action, returning only after the action is completed.

Parameters

- **bucket** (`b2sdk.bucket.Bucket`) – a Bucket object
- **reporter** – a place to report errors

`do_report(bucket, reporter)`

Report the downloading action performed.

Parameters

- **bucket** (`b2sdk.bucket.Bucket`) – a Bucket object
- **reporter** – a place to report errors

get_bytes ()

Return file size

Return type `int`

class `b2sdk.sync.action.B2HideAction` (*relative_name, b2_file_name*)

Bases: `b2sdk.sync.action.AbstractAction`

__init__ (*relative_name, b2_file_name*)

Parameters

- **relative_name** (*str*) – a relative file name
- **b2_file_name** (*str*) – a name of a remote file

do_action (*bucket, reporter*)

Performs the hiding action, returning only after the action is completed.

Parameters

- **bucket** (`b2sdk.bucket.Bucket`) – a Bucket object
- **reporter** – a place to report errors

do_report (*bucket, reporter*)

Report the hiding action performed.

Parameters

- **bucket** (`b2sdk.bucket.Bucket`) – a Bucket object
- **reporter** – a place to report errors

get_bytes ()

Return file size

Returns always zero

Return type `int`

class `b2sdk.sync.action.B2UploadAction` (*local_full_path, relative_name, b2_file_name, mod_time_millis, size*)

Bases: `b2sdk.sync.action.AbstractAction`

File uploading action

__init__ (*local_full_path, relative_name, b2_file_name, mod_time_millis, size*)

Parameters

- **local_full_path** (*str*) – a local file path
- **relative_name** (*str*) – a relative file name
- **b2_file_name** (*str*) – a name of a new remote file
- **mod_time_millis** (*int*) – file modification time in milliseconds
- **size** (*int*) – a file size

do_action (*bucket, reporter*)

Performs the uploading action, returning only after the action is completed.

Parameters

- **bucket** (`b2sdk.bucket.Bucket`) – a Bucket object
- **reporter** – a place to report errors

do_report (*bucket, reporter*)

Report the uploading action performed.

Parameters

- **bucket** (`b2sdk.bucket.Bucket`) – a Bucket object
- **reporter** – a place to report errors

get_bytes ()

Return file size

Return type `int`

class `b2sdk.sync.action.LocalDeleteAction` (*relative_name, full_path*)

Bases: `b2sdk.sync.action.AbstractAction`

__init__ (*relative_name, full_path*)

Parameters

- **relative_name** (*str*) – a relative file name
- **full_path** – a full local path

Type `str`

do_action (*bucket, reporter*)

Performs the deleting of a local file action, returning only after the action is completed.

Parameters

- **bucket** (`b2sdk.bucket.Bucket`) – a Bucket object
- **reporter** – a place to report errors

do_report (*bucket, reporter*)

Report the deleting of a local file action performed.

Parameters

- **bucket** (`b2sdk.bucket.Bucket`) – a Bucket object
- **reporter** – a place to report errors

get_bytes ()

Return file size

Returns always zero

Return type `int`

b2sdk.sync.exception – Sync exceptions

exception `b2sdk.sync.exception.EnvironmentEncodingError` (*filename, encoding*)

Bases: `b2sdk.exception.B2Error`

Raised when a file name can not be decoded with system encoding

__init__ (*filename, encoding*)

Parameters

- **filename** (*str*, *bytes*) – an encoded file name
- **encoding** (*str*) – file name encoding

b2sdk.sync.file – File

class b2sdk.sync.file.**File** (*name*, *versions*)

Bases: *object*

Holds information about one file in a folder.

The name is relative to the folder in all cases.

Files that have multiple versions (which only happens in B2, not in local folders) include information about all of the versions, most recent first.

__init__ (*name*, *versions*)

Parameters

- **name** (*str*) – a relative file name
- **versions** (*list*) – a list of file versions

latest_version ()

Return the latest file version

class b2sdk.sync.file.**FileVersion** (*id_*, *file_name*, *mod_time*, *action*, *size*)

Bases: *object*

Holds information about one version of a file:

__init__ (*id_*, *file_name*, *mod_time*, *action*, *size*)

Parameters

- **id** (*str*) – the B2 file id, or the local full path name
- **file_name** (*str*) – a relative file name
- **mod_time** (*int*) – modification time, in milliseconds, to avoid rounding issues with millisecond times from B2
- **action** (*str*) – “hide” or “upload” (never “start”)
- **size** (*int*) – a file size

b2sdk.sync.folder_parser – Folder parser

b2sdk.sync.folder_parser.**parse_sync_folder** (*folder_name*, *api*)

Takes either a local path, or a B2 path, and returns a Folder object for it.

B2 paths look like: b2://bucketName/path/name. The ‘//’ is optional, because the previous sync command didn’t use it.

Anything else is treated like a local folder.

Parameters

- **folder_name** (*str*) – a name of the folder, either local or remote
- **api** (*b2sdk.api.B2Api*) – an API object

b2sdk.sync.folder – Folder**class** b2sdk.sync.folder.**AbstractFolder**Bases: `object`

Interface to a folder full of files, which might be a B2 bucket, a virtual folder in a B2 bucket, or a directory on a local file system.

Files in B2 may have multiple versions, while files in local folders have just one.

abstract `all_files` (*reporter, policies_manager*)

Returns an iterator over all of the files in the folder, in the order that B2 uses.

It also performs filtering using policies manager.

No matter what the folder separator on the local file system is, “/” is used in the returned file names.

If a file is found, but does not exist (for example due to a broken symlink or a race), reporter will be informed about each such problem.

Parameters

- **reporter** – a place to report errors
- **policies_manager** – a policies manager object

abstract `folder_type` ()

Returns one of: ‘b2’, ‘local’

Return type `str`**abstract** `make_full_path` (*file_name*)

Only for local folders, returns the full path to the file.

Parameters `file_name` (*str*) – a file name**class** b2sdk.sync.folder.**B2Folder** (*bucket_name, folder_name, api*)Bases: `b2sdk.sync.folder.AbstractFolder`

Folder interface to b2.

`__init__` (*bucket_name, folder_name, api*)**Parameters**

- **bucket_name** (*str*) – a name of the bucket
- **folder_name** (*str*) – a folder name
- **api** (`b2sdk.api.B2Api`) – an API object

all_files (*reporter, policies_manager=<b2sdk.sync.scan_policies.ScanPoliciesManager object>*)

Yield all files

Parameters

- **reporter** – a place to report errors
- **policies_manager** – a policies manager object, default is `DEFAULT_SCAN_MANAGER`

folder_type ()

Return folder type

Return type `str`

make_full_path (*file_name*)

Make an absolute path from a file name

Parameters **file_name** (*str*) – a file name

class `b2sdk.sync.folder.LocalFolder` (*root*)

Bases: `b2sdk.sync.folder.AbstractFolder`

Folder interface to a directory on the local machine.

__init__ (*root*)

Initializes a new folder.

Parameters **root** (*str*) – Path to the root of the local folder. Must be unicode.

all_files (*reporter, policies_manager=<b2sdk.sync.scan_policies.ScanPoliciesManager object>*)

Yield all files

Parameters

- **reporter** – a place to report errors
- **policies_manager** – a policy manager object, default is DE-FAULT_SCAN_MANAGER

ensure_non_empty ()

Makes sure that the directory exists and is non-empty.

ensure_present ()

Makes sure that the directory exists.

folder_type ()

Return folder type

Return type *str*

make_full_path (*file_name*)

Convert a file name into an absolute path

Parameters **file_name** (*str*) – a file name

`b2sdk.sync.folder.join_b2_path` (*b2_dir, b2_name*)

Like `os.path.join`, but for B2 file names where the root directory is called ‘.’.

Parameters

- **b2_dir** (*str*) – a directory path
- **b2_name** (*str*) – a file name

b2sdk.sync.policy_manager – Policy manager

class `b2sdk.sync.policy_manager.SyncPolicyManager`

Bases: `object`

Policy manager, implements a logic to get a correct policy class and create a policy object based on various parameters

__init__ ()

Initialize self. See `help(type(self))` for accurate signature.

get_policy (*sync_type, source_file, source_folder, dest_file, dest_folder, now_millis, args*)

Return policy object

Parameters

- **sync_type** (*str*) – synchronization type
- **source_file** (*str*) – source file name
- **source_folder** (*str*) – a source folder path
- **dest_file** (*str*) – destination file name
- **dest_folder** (*str*) – a destination folder path
- **now_millis** (*int*) – current time in milliseconds
- **args** – an object which holds command line arguments

Returns a policy object

get_policy_class (*sync_type*, *args*)

Get policy class by a given sync type

Parameters

- **sync_type** (*str*) – synchronization type
- **args** – an object which holds command line arguments

Returns a policy class

b2sdk.sync.policy – Policy

class b2sdk.sync.policy.**AbstractFileSyncPolicy** (*source_file*, *source_folder*, *dest_file*,
dest_folder, *now_millis*, *args*)

Bases: `object`

Abstract policy class

DESTINATION_PREFIX = `NotImplemented`

SOURCE_PREFIX = `NotImplemented`

__init__ (*source_file*, *source_folder*, *dest_file*, *dest_folder*, *now_millis*, *args*)

Parameters

- **source_file** (`b2sdk.sync.file.File`) – source file object
- **source_folder** (`b2sdk.sync.folder.AbstractFolder`) – source folder object
- **dest_file** (`b2sdk.sync.file.File`) – destination file object
- **dest_folder** (`b2sdk.sync.folder.AbstractFolder`) – destination folder object
- **now_millis** (*int*) – current time in milliseconds
- **args** – an object which holds command line arguments

classmethod **files_are_different** (*source_file*, *dest_file*, *args*)

Compare two files and determine if the the destination file should be replaced by the source file.

Parameters

- **source_file** (`b2sdk.sync.file.File`) – source file object
- **dest_file** (`b2sdk.sync.file.File`) – destination file object
- **args** – an object which holds command line arguments

Return type `bool`

get_all_actions()

Yield file actions.

class `b2sdk.sync.policy.DownAndDeletePolicy` (*source_file, source_folder, dest_file, dest_folder, now_millis, args*)

Bases: `b2sdk.sync.policy.DownPolicy`

File is synced down (from the cloud to disk) and the delete flag is SET

class `b2sdk.sync.policy.DownAndKeepDaysPolicy` (*source_file, source_folder, dest_file, dest_folder, now_millis, args*)

Bases: `b2sdk.sync.policy.DownPolicy`

File is synced down (from the cloud to disk) and the keepDays flag is SET

class `b2sdk.sync.policy.DownPolicy` (*source_file, source_folder, dest_file, dest_folder, now_millis, args*)

Bases: `b2sdk.sync.policy.AbstractFileSyncPolicy`

File is synced down (from the cloud to disk)

DESTINATION_PREFIX = 'local:/'

SOURCE_PREFIX = 'b2:/'

class `b2sdk.sync.policy.UpAndDeletePolicy` (*source_file, source_folder, dest_file, dest_folder, now_millis, args*)

Bases: `b2sdk.sync.policy.UpPolicy`

File is synced up (from disk to the cloud) and the delete flag is SET

class `b2sdk.sync.policy.UpAndKeepDaysPolicy` (*source_file, source_folder, dest_file, dest_folder, now_millis, args*)

Bases: `b2sdk.sync.policy.UpPolicy`

File is synced up (from disk to the cloud) and the keepDays flag is SET

class `b2sdk.sync.policy.UpPolicy` (*source_file, source_folder, dest_file, dest_folder, now_millis, args*)

Bases: `b2sdk.sync.policy.AbstractFileSyncPolicy`

File is synced up (from disk the cloud)

DESTINATION_PREFIX = 'b2:/'

SOURCE_PREFIX = 'local:/'

`b2sdk.sync.policy.make_b2_delete_actions` (*source_file, dest_file, dest_folder, transferred*)

Creates the actions to delete files stored on B2, which are not present locally.

Parameters

- **source_file** (`b2sdk.sync.file.File`) – source file object
- **dest_file** (`b2sdk.sync.file.File`) – destination file object
- **dest_folder** (`b2sdk.sync.folder.AbstractFolder`) – destination folder
- **transferred** (`bool`) – if True, file has been transferred, False otherwise

`b2sdk.sync.policy.make_b2_delete_note` (*version, index, transferred*)

Create a note message for delete action

Parameters

- **version** – an object which contains file version info

- **index** (*int*) – file version index
- **transferred** (*bool*) – if True, file has been transferred, False otherwise

`b2sdk.sync.policy.make_b2_keep_days_actions` (*source_file, dest_file, dest_folder, transferred, keep_days, now_millis*)

Creates the actions to hide or delete existing versions of a file stored in b2.

When keepDays is set, all files that were visible any time from keepDays ago until now must be kept. If versions were uploaded 5 days ago, 15 days ago, and 25 days ago, and the keepDays is 10, only the 25-day old version can be deleted. The 15 day-old version was visible 10 days ago.

Parameters

- **source_file** (`b2sdk.sync.file.File`) – source file object
- **dest_file** (`b2sdk.sync.file.File`) – destination file object
- **dest_folder** (`b2sdk.sync.folder.AbstractFolder`) – destination folder object
- **transferred** (*bool*) – if True, file has been transferred, False otherwise
- **keep_days** (*int*) – how many days to keep a file
- **now_millis** (*int*) – current time in milliseconds

`b2sdk.sync.scan_policies` – Scan policies

class `b2sdk.sync.scan_policies.RegexSet` (*regex_iterable*)

Bases: `object`

Holds a (possibly empty) set of regular expressions, and knows how to check whether a string matches any of them.

__init__ (*regex_iterable*)

Parameters *regex_iterable* – an iterable which yields regexes

matches (*s*)

Check whether a string matches any of regular expressions

Parameters *s* (*str*) – a string to check

Return type `bool`

class `b2sdk.sync.scan_policies.ScanPoliciesManager` (*exclude_dir_regexes=(), exclude_file_regexes=(), include_file_regexes=(), exclude_all_symlinks=False*)

Bases: `object`

Policy object used when scanning folders for syncing, used to decide which files to include in the list of files to be synced.

Code that scans through files should at least use `should_exclude_file()` to decide whether each file should be included; it will check include/exclude patterns for file names, as well as patterns for excluding directories.

Code that scans may optionally use `should_exclude_directory()` to test whether it can skip a directory completely and not bother listing the files and sub-directories in it.

__init__ (*exclude_dir_regexes=(), exclude_file_regexes=(), include_file_regexes=(), exclude_all_symlinks=False*)

Parameters

- **exclude_dir_regexes** (*tuple*) – a tuple of regexes to exclude directories
- **exclude_file_regexes** (*tuple*) – a tuple of regexes to exclude files
- **include_file_regexes** (*tuple*) – a tuple of regexes to include files
- **exclude_all_symlinks** (*bool*) – if True, exclude all symlinks

should_exclude_directory (*dir_path*)

Given the full path of a directory, should all of the files in it be excluded from the scan?

Parameters **dir_path** (*str*) – The path of the directory, relative to the root directory being scanned. The path will never end in '/'.

Returns True if excluded.

should_exclude_file (*file_path*)

Given the full path of a file, should it be excluded from the scan?

Parameters **file_path** – The path of the file, relative to the root directory being scanned.

Type *str*

Returns True if excluded.

Return type *bool*

b2sdk.sync.scan_policies.convert_dir_regex_to_dir_prefix_regex (*dir_regex*)

The patterns used to match directory names (and file names) are allowed to match a prefix of the name. This 'feature' was unintentional, but is being retained for compatibility.

This means that a regex that matches a directory name can't be used directly to match against a file name and test whether the file should be excluded because it matches the directory.

The pattern 'photos' will match directory names 'photos' and 'photos2', and should exclude files 'photos/kitten.jpg', and 'photos2/puppy.jpg'. It should not exclude 'photos.txt', because there is no directory name that matches.

On the other hand, the pattern 'photos\$' should match 'photos/kitten.jpg', but not 'photos2/puppy.jpg', nor 'photos.txt'

If the original regex is valid, there are only two cases to consider: either the regex ends in '\$' or does not.

Parameters **dir_regex** (*str*) – a regular expression string or literal

b2sdk.sync.sync – Sync

b2sdk.sync.sync.count_files (*local_folder, reporter*)

Counts all of the files in a local folder.

Parameters

- **local_folder** (*b2sdk.sync.folder.AbstractFolder*) – a folder object.
- **reporter** – reporter object

b2sdk.sync.sync.make_file_sync_actions (*sync_type, source_file, dest_file, source_folder, dest_folder, args, now_millis*)

Yields the sequence of actions needed to sync the two files

Parameters

- **sync_type** (*str*) – synchronization type
- **source_file** – source file object

- **dest_file** (`b2sdk.sync.file.File`) – destination file object
- **source_folder** (`b2sdk.sync.folder.AbstractFolder`) – a source folder object
- **dest_folder** (`b2sdk.sync.folder.AbstractFolder`) – a destination folder object
- **args** – an object which holds command line arguments
- **now_millis** (`int`) – current time in milliseconds

```
b2sdk.sync.sync.make_folder_sync_actions (source_folder,      dest_folder,      args,
                                         now_millis,        reporter,        poli-
                                         cies_manager=<b2sdk.sync.scan_policies.ScanPoliciesManager
                                         object>)
```

Yields a sequence of actions that will sync the destination folder to the source folder.

Parameters

- **source_folder** (`b2sdk.sync.folder.AbstractFolder`) – source folder object
- **dest_folder** (`b2sdk.sync.folder.AbstractFolder`) – destination folder object
- **args** – an object which holds command line arguments
- **now_millis** (`int`) – current time in milliseconds
- **reporter** – reporter object
- **policies_manager** – policies manager object

```
b2sdk.sync.sync.next_or_none (iterator)
```

Returns the next item from the iterator, or None if there are no more.

```
b2sdk.sync.sync.sync_folders (source_folder,      dest_folder,      args,      now_millis,
                             stdout,            no_progress,      max_workers,      poli-
                             cies_manager=<b2sdk.sync.scan_policies.ScanPoliciesManager
                             object>, dry_run=False, allow_empty_source=False)
```

Syncs two folders. Always ensures that every file in the source is also in the destination. Deletes any file versions in the destination older than `history_days`.

Parameters

- **source_folder** (`b2sdk.sync.folder.AbstractFolder`) – source folder object
- **dest_folder** (`b2sdk.sync.folder.AbstractFolder`) – destination folder object
- **args** – an object which holds command line arguments
- **now_millis** (`int`) – current time in milliseconds
- **stdout** – standard output file object
- **no_progress** (`bool`) – if True, do not show progress
- **max_workers** (`int`) – max number of workers
- **policies_manager** – policies manager object
- **dry_run** (`bool`) –
- **allow_empty_source** (`bool`) – if True, do not check whether source folder is empty

```
b2sdk.sync.sync.zip_folders(folder_a, folder_b, reporter, policies_manager=<b2sdk.sync.scan_policies.ScanPoliciesManager object>)
```

An iterator over all of the files in the union of two folders, matching file names.

Each item is a pair (file_a, file_b) with the corresponding file in both folders. Either file (but not both) will be None if the file is in only one folder.

Parameters

- **folder_a** (`b2sdk.sync.folder.AbstractFolder`) – first folder object.
- **folder_b** (`b2sdk.sync.folder.AbstractFolder`) – second folder object.
- **reporter** – reporter object
- **policies_manager** – policies manager object

Returns yields two element tuples

4.2 Internal API

Warning: See *Internal interface* chapter to learn when and how to safely use the Internal API

4.2.1 b2sdk.session – B2Session class

```
class b2sdk.session.B2Session(api, raw_api)
    Bases: object
```

A *magic* facade that supplies the correct `api_url` and `account_auth_token` to methods of underlying `raw_api` and reauthorizes if necessary

```
__init__(api, raw_api)
    Initialize self. See help(type(self)) for accurate signature.
```

4.2.2 b2sdk.raw_api – B2 raw api wrapper

```
class b2sdk.raw_api.AbstractRawApi
    Bases: object
```

Direct access to the B2 web apis.

```
abstract cancel_large_file(api_url, account_auth_token, file_id)
abstract copy_file(api_url, account_auth_token, source_file_id, new_file_name,
                  bytes_range=None, metadata_directive=None, content_type=None,
                  file_info=None, destination_bucket_id=None)
abstract delete_bucket(api_url, account_auth_token, account_id, bucket_id)
abstract delete_file_version(api_url, account_auth_token, file_id, file_name)
abstract finish_large_file(api_url, account_auth_token, file_id, part_sha1_array)
get_download_url_by_id(download_url, account_auth_token, file_id)
get_download_url_by_name(download_url, account_auth_token, bucket_name, file_name)
abstract get_upload_part_url(api_url, account_auth_token, file_id)
```

```
abstract hide_file (api_url, account_auth_token, bucket_id, file_name)  
abstract list_parts (api_url, account_auth_token, file_id, start_part_number, max_part_count)  
abstract list_unfinished_large_files (api_url, account_auth_token, bucket_id,  
                                     start_file_id=None, max_file_count=None)  
abstract start_large_file (api_url, account_auth_token, bucket_id, file_name, content_type,  
                           file_info)  
abstract update_bucket (api_url, account_auth_token, account_id, bucket_id,  
                        bucket_type=None, bucket_info=None, cors_rules=None, lifecy-  
                        cle_rules=None, if_revision_is=None)  
abstract upload_part (upload_url, upload_auth_token, part_number, content_length, sha1_sum,  
                      input_stream)
```

```
class b2sdk.raw_api.B2RawApi (b2_http)
```

```
    Bases: b2sdk.raw_api.AbstractRawApi
```

Provides access to the B2 web APIs, exactly as they are provided by b2.

Requires that you provide all necessary URLs and auth tokens for each call.

Each API call decodes the returned JSON and returns a dict.

For details on what each method does, see the B2 docs: <https://www.backblaze.com/b2/docs/>

This class is intended to be a super-simple, very thin layer on top of the HTTP calls. It can be mocked-out for testing higher layers. And this class can be tested by exercising each call just once, which is relatively quick.

All public methods of this class except `authorize_account` shall accept `api_url` and `account_info` as first two positional arguments. This is needed for `B2Session` magic.

```
    __init__ (b2_http)
```

Initialize self. See `help(type(self))` for accurate signature.

```
    authorize_account (realm_url, application_key_id, application_key)
```

```
    cancel_large_file (api_url, account_auth_token, file_id)
```

```
    check_b2_filename (filename)
```

Raise an appropriate exception with details if the filename is unusable.

See <https://www.backblaze.com/b2/docs/files.html> for the rules.

Parameters filename – A proposed filename in unicode.

Returns `None` if the filename is usable.

```
    copy_file (api_url, account_auth_token, source_file_id, new_file_name, bytes_range=None, meta-  
             data_directive=None, content_type=None, file_info=None, destination_bucket_id=None)
```

```
    create_bucket (api_url, account_auth_token, account_id, bucket_name, bucket_type,  
                 bucket_info=None, cors_rules=None, lifecycle_rules=None)
```

```
    create_key (api_url, account_auth_token, account_id, capabilities, key_name,  
              valid_duration_seconds, bucket_id, name_prefix)
```

```
    delete_bucket (api_url, account_auth_token, account_id, bucket_id)
```

```
    delete_file_version (api_url, account_auth_token, file_id, file_name)
```

```
    delete_key (api_url, account_auth_token, application_key_id)
```

```
    download_file_from_url (_, account_auth_token_or_none, url, range_=None)
```

Issues a streaming request for download of a file, potentially authorized.

Parameters

- `_` – unused (caused by B2Session magic)
- `account_auth_token_or_none` – an optional account auth token to pass in
- `url` – The full URL to download from
- `range` – two-element tuple for http Range header

Returns b2_http response

```

finish_large_file (api_url, account_auth_token, file_id, part_sha1_array)
get_download_authorization (api_url, account_auth_token, bucket_id, file_name_prefix,
                             valid_duration_in_seconds)
get_file_info (api_url, account_auth_token, file_id)
get_upload_part_url (api_url, account_auth_token, file_id)
get_upload_url (api_url, account_auth_token, bucket_id)
hide_file (api_url, account_auth_token, bucket_id, file_name)
list_buckets (api_url, account_auth_token, account_id, bucket_id=None, bucket_name=None)
list_file_names (api_url, account_auth_token, bucket_id, start_file_name=None,
                  max_file_count=None, prefix=None)
list_file_versions (api_url, account_auth_token, bucket_id, start_file_name=None,
                      start_file_id=None, max_file_count=None, prefix=None)
list_keys (api_url, account_auth_token, account_id, max_key_count=None,
             start_application_key_id=None)
list_parts (api_url, account_auth_token, file_id, start_part_number, max_part_count)
list_unfinished_large_files (api_url, account_auth_token, bucket_id, start_file_id=None,
                              max_file_count=None)
start_large_file (api_url, account_auth_token, bucket_id, file_name, content_type, file_info)
unprintable_to_hex (string)

```

Replace unprintable chars in string with a hex representation.

Parameters `string` – An arbitrary string, possibly with unprintable characters.

Returns The string, with unprintable characters changed to hex (e.g., “”)

```

update_bucket (api_url, account_auth_token, account_id, bucket_id, bucket_type=None,
                bucket_info=None, cors_rules=None, lifecycle_rules=None, if_revision_is=None)
upload_file (upload_url, upload_auth_token, file_name, content_length, content_type, content_sha1,
              file_infos, data_stream)

```

Uploads one small file to b2.

Parameters

- `upload_url` – The upload_url from b2_authorize_account
- `upload_auth_token` – The auth token from b2_authorize_account
- `file_name` – The name of the B2 file
- `content_length` – Number of bytes in the file.
- `content_type` – MIME type.
- `content_sha1` – Hex SHA1 of the contents of the file
- `file_infos` – Extra file info to upload

- **data_stream** – A file like object from which the contents of the file can be read.

Returns

upload_part (*upload_url, upload_auth_token, part_number, content_length, content_shal, data_stream*)

class b2sdk.raw_api.**MetadataDirectiveMode**

Bases: `enum.Enum`

An enumeration.

COPY = 401

REPLACE = 402

b2sdk.raw_api.**test_raw_api**()

Exercises the code in B2RawApi by making each call once, just to make sure the parameters are passed in, and the result is passed back.

The goal is to be a complete test of B2RawApi, so the tests for the rest of the code can use the simulator.

Prints to stdout if things go wrong.

Returns 0 on success, non-zero on failure.

b2sdk.raw_api.**test_raw_api_helper**(*raw_api*)

Tries each of the calls to the raw api. Raises an except if anything goes wrong.

This uses a Backblaze account that is just for this test. The account uses the free level of service, which should be enough to run this test a reasonable number of times each day. If somebody abuses the account for other things, this test will break and we'll have to do something about it.

4.2.3 b2sdk.b2http – B2 http client wrapper

class b2sdk.b2http.**B2Http** (*requests_module=None, install_clock_skew_hook=True*)

Bases: `object`

A wrapper for the requests module. Provides the operations needed to access B2, and handles retrying when the returned status is 503 Service Unavailable, 429 Too Many Requests etc

The operations supported are:

- **post_json_return_json**
- **post_content_return_json**
- **get_content**

The methods that return JSON either return a Python dict or raise a subclass of B2Error. They can be used like this:

```
try:
    response_dict = b2_http.post_json_return_json(url, headers, params)
    ...
except B2Error as e:
    ...
```

TIMEOUT = 130

__init__ (*requests_module=None, install_clock_skew_hook=True*)

Initialize with a reference to the requests module, which makes it easy to mock for testing.

Parameters

- **requests_module** – a reference to requests module
- **install_clock_skew_hook** (*bool*) – if True, install a clock skew hook

add_callback (*callback*)

Adds a callback that inherits from HttpCallback.

Parameters **callback** (*callable*) – a callback to be added to a chain

get_content (*url, headers, try_count=5*)

Fetches content from a URL.

Use like this:

```
try:
    with b2_http.get_content(url, headers) as response:
        for byte_data in response.iter_content(chunk_size=1024):
            ...
except B2Error as e:
    ...
```

The response object is only guarantee to have:

- headers
- iter_content()

Parameters

- **url** (*str*) – URL to call
- **headers** (*dict*) – headers to send
- **try_count** (*int*) – a number of retries

Returns Context manager that returns an object that supports iter_content()

post_content_return_json (*url, headers, data, try_count=5, post_params=None*)

Use like this:

```
try:
    response_dict = b2_http.post_content_return_json(url, headers, data)
    ...
except B2Error as e:
    ...
```

Parameters

- **url** (*str*) – URL to call
- **headers** (*dict*) – headers to send.
- **data** – bytes (Python 3) or str (Python 2), or a file-like object, to send

Returns a dict that is the decoded JSON

Return type *dict*

post_json_return_json (*url, headers, params, try_count=5*)

Use like this:

```
try:
    response_dict = b2_http.post_json_return_json(url, headers, params)
    ...
except B2Error as e:
    ...
```

Parameters

- **url** (*str*) – URL to call
- **headers** (*dict*) – headers to send.
- **params** (*dict*) – a dict that will be converted to JSON

Returns the decoded JSON document

Return type *dict*

class b2sdk.b2http.**ClockSkewHook**

Bases: *b2sdk.b2http.HttpCallback*

post_request (*method, url, headers, response*)

Raises an exception if the clock in the server is too different from the clock on the local host.

The Date header contains a string that looks like: “Fri, 16 Dec 2016 20:52:30 GMT”.

Parameters

- **method** (*str*) – one of: ‘POST’, ‘GET’, etc.
- **url** (*str*) – the URL that will be used.
- **headers** (*dict*) – the header sent with the request.
- **response** – a response object from the requests library.

class b2sdk.b2http.**HttpCallback**

Bases: *object*

A callback object that does nothing. Override *pre_request* and/or *post_request* as desired.

post_request (*method, url, headers, response*)

Called after processing an HTTP request. Should not raise an exception.

Raises an exception if this request should be treated as failing. The exception raised must inherit from *B2HttpCallbackPostRequestException*.

Parameters

- **method** (*str*) – one of: ‘POST’, ‘GET’, etc.
- **url** (*str*) – the URL that will be used.
- **headers** (*dict*) – the header sent with the request.
- **response** – a response object from the requests library.

pre_request (*method, url, headers*)

Called before processing an HTTP request.

Raises an exception if this request should not be processed. The exception raised must inherit from *B2HttpCallbackPreRequestException*.

Parameters

- **method** (*str*) – str, One of: ‘POST’, ‘GET’, etc.

- **url** (*str*) – The URL that will be used.
- **headers** (*dict*) – The header sent with the request.

class b2sdk.b2http.**ResponseContextManager** (*response*)

Bases: `object`

Context manager that closes a requests.Response when done.

__init__ (*response*)

Initialize self. See help(type(self)) for accurate signature.

b2sdk.b2http.**test_http** ()

Runs a few tests on error diagnosis.

This test takes a while to run, and is not used in the automated tests during building. Run the test by hand to exercise the code. Be sure to run in both Python 2 and Python 3.

4.2.4 b2sdk.utils – Utils

class b2sdk.utils.**B2TraceMeta**

Bases: `logfury.v0_1.meta.DefaultTraceMeta`

Traces all public method calls, except for ones with names that begin with *get_*

class b2sdk.utils.**B2TraceMetaAbstract**

Bases: `logfury.v0_1.meta.DefaultTraceAbstractMeta`

Default class for tracers, to be set as a metaclass for abstract base classes

class b2sdk.utils.**BytesIoContextManager** (*byte_data*)

Bases: `object`

A simple wrapper for a BytesIO that makes it look like a file-like object that can be a context manager.

__init__ (*byte_data*)

Parameters *bytes_data* – a byte stream

class b2sdk.utils.**TempDir**

Bases: `object`

Context manager that creates and destroys a temporary directory.

b2sdk.utils.**b2_url_decode** (*s*)

Decodes a Unicode string returned from B2 in an HTTP header.

Parameters *s* (*str*) – a unicode string to decode

Returns a Python unicode string.

Return type `str`

b2sdk.utils.**b2_url_encode** (*s*)

URL-encodes a unicode string to be sent to B2 in an HTTP header.

Parameters *s* (*str*) – a unicode string to encode

Returns URL-encoded string

Return type `str`

b2sdk.utils.**camelcase_to_underscore** (*input_*)

Convert camel cased string to string with underscores

Parameters *input* (*str*) – an input string

Returns string with underscores

Return type `str`

`b2sdk.utils.choose_part_ranges` (*content_length*, *minimum_part_size*)

Returns a list of (offset, length) for the parts of a large file.

Parameters

- **content_length** (*int*) – content length value
- **minimum_part_size** (*int*) – a minimum file part size

Return type `list`

`b2sdk.utils.fix_windows_path_limit` (*path*)

Prefix paths when running on Windows to overcome 260 character path length limit See [https://msdn.microsoft.com/en-us/library/windows/desktop/aa365247\(v=vs.85\).aspx#maxpath](https://msdn.microsoft.com/en-us/library/windows/desktop/aa365247(v=vs.85).aspx#maxpath)

Parameters **path** (*str*) – a path to prefix

Returns a prefixed path

Return type `str`

`b2sdk.utils.format_and_scale_fraction` (*numerator*, *denominator*, *unit*)

Picks a good scale for representing a fraction, and formats it.

Parameters

- **numerator** (*int*) – a numerator of a fraction
- **denominator** (*int*) – a denominator of a fraction
- **unit** (*str*) – an arbitrary unit name

Returns scaled and formatted fraction

Return type `str`

`b2sdk.utils.format_and_scale_number` (*x*, *unit*)

Picks a good scale for representing a number and formats it.

Parameters

- **x** (*int*) – a number
- **unit** (*str*) – an arbitrary unit name

Returns scaled and formatted number

Return type `str`

`b2sdk.utils.hex_sha1_of_bytes` (*data*)

Returns the 40-character hex SHA1 checksum of the data

Parameters **data** (*bytes*) – an array of bytes

Return type `str`

`b2sdk.utils.hex_sha1_of_stream` (*input_stream*, *content_length*)

Returns the 40-character hex SHA1 checksum of the first *content_length* bytes in the input stream.

Parameters

- **input_stream** – stream object, which exposes `read()` method
- **content_length** (*int*) – expected length of the stream

Return type `str`

`b2sdk.utils.interruptible_get_result(future)`

Waits for the result of a future in a way that can be interrupted by a KeyboardInterrupt.

This is not necessary in Python 3, but is needed for Python 2.

Parameters `future` (*Future*) – a future to get result of

`b2sdk.utils.is_file_readable(local_path, reporter=None)`

Check if the local file has read permissions

Parameters

- `local_path` (*str*) – a file path
- `reporter` – reporter object to put errors on

Return type `bool`

`b2sdk.utils.validate_b2_file_name(name)`

Raises a ValueError if the name is not a valid B2 file name.

Parameters `name` (*str*) – a string to check

4.2.5 `b2sdk.transferer.abstract`

class `b2sdk.transferer.abstract.AbstractDownloader` (*force_chunk_size=None*,
min_chunk_size=None,
max_chunk_size=None)

Bases: `object`

__init__ (*force_chunk_size=None*, *min_chunk_size=None*, *max_chunk_size=None*)

Initialize self. See help(type(self)) for accurate signature.

abstract download (*file*, *response*, *metadata*, *session*)

@returns (bytes_read, actual_sha1)

abstract is_suitable (*metadata*, *progress_listener*)

analyzes metadata (possibly against options passed earlier to constructor to find out whether the given download request should be handled by this downloader)

4.2.6 `b2sdk.transferer.file_metadata`

class `b2sdk.transferer.file_metadata.FileMetadata` (*file_id*, *file_name*, *content_type*,
content_length, *content_sha1*,
file_info)

Bases: `object`

holds information about a file which is being downloaded

__init__ (*file_id*, *file_name*, *content_type*, *content_length*, *content_sha1*, *file_info*)

Initialize self. See help(type(self)) for accurate signature.

as_info_dict ()

content_length

content_sha1

content_type

file_id

```
file_info
file_name
classmethod from_response(response)
```

4.2.7 b2sdk.transferer.parallel

```
class b2sdk.transferer.parallel.AbstractDownloaderThread(session, writer,
                                                         part_to_download,
                                                         chunk_size)
```

Bases: `threading.Thread`

```
__init__(session, writer, part_to_download, chunk_size)
```

Parameters

- **session** – raw_api wrapper
- **writer** – where to write data
- **part_to_download** – PartToDownload object
- **chunk_size** – internal buffer size to use for writing and hashing

```
abstract run()
```

Method representing the thread's activity.

You may override this method in a subclass. The standard `run()` method invokes the callable object passed to the object's constructor as the target argument, if any, with sequential and keyword arguments taken from the `args` and `kwargs` arguments, respectively.

```
class b2sdk.transferer.parallel.FirstPartDownloaderThread(response, hasher, *args,
                                                         **kwargs)
```

Bases: `b2sdk.transferer.parallel.AbstractDownloaderThread`

```
__init__(response, hasher, *args, **kwargs)
```

Parameters

- **response** – response of the original GET call
- **hasher** – hasher object to feed to as the stream is written

```
run()
```

Method representing the thread's activity.

You may override this method in a subclass. The standard `run()` method invokes the callable object passed to the object's constructor as the target argument, if any, with sequential and keyword arguments taken from the `args` and `kwargs` arguments, respectively.

```
class b2sdk.transferer.parallel.NonHashingDownloaderThread(url, *args, **kwargs)
```

Bases: `b2sdk.transferer.parallel.AbstractDownloaderThread`

```
__init__(url, *args, **kwargs)
```

Parameters **url** – url of the target file

```
run()
```

Method representing the thread's activity.

You may override this method in a subclass. The standard `run()` method invokes the callable object passed to the object's constructor as the target argument, if any, with sequential and keyword arguments taken from the `args` and `kwargs` arguments, respectively.

```
class b2sdk.transferer.parallel.ParallelDownloader (max_streams, min_part_size,  
                                                    *args, **kwargs)
```

Bases: *b2sdk.transferer.abstract.AbstractDownloader*

FINISH_HASHING_BUFFER_SIZE = 1048576

```
__init__ (max_streams, min_part_size, *args, **kwargs)
```

Parameters

- **max_streams** – maximum number of simultaneous streams
- **min_part_size** – minimum amount of data a single stream will retrieve, in bytes

```
download (file, response, metadata, session)
```

Downloads a file from given url using parallel download sessions and stores it in the given download_destination.

Parameters

- **file** – an opened file-like object to write to
- **response** – The response of the first request made to the cloud service with download intent

Returns

```
is_suitable (metadata, progress_listener)
```

analyzes metadata (possibly against options passed earlier to constructor to find out whether the given download request should be handled by this downloader)

```
class b2sdk.transferer.parallel.PartToDownload (cloud_range, local_range)
```

Bases: *object*

Holds the range of a file to download, and the range of the local file where it should be stored.

```
__init__ (cloud_range, local_range)
```

Initialize self. See help(type(self)) for accurate signature.

```
class b2sdk.transferer.parallel.WriterThread (file)
```

Bases: *threading.Thread*

```
__init__ (file)
```

This constructor should always be called with keyword arguments. Arguments are:

group should be None; reserved for future extension when a ThreadGroup class is implemented.

target is the callable object to be invoked by the run() method. Defaults to None, meaning nothing is called.

name is the thread name. By default, a unique name is constructed of the form “Thread-N” where N is a small decimal number.

args is the argument tuple for the target invocation. Defaults to ().

kwargs is a dictionary of keyword arguments for the target invocation. Defaults to {}.

If a subclass overrides the constructor, it must make sure to invoke the base class constructor (Thread.__init__()) before doing anything else to the thread.

```
run ()
```

Method representing the thread’s activity.

You may override this method in a subclass. The standard run() method invokes the callable object passed to the object’s constructor as the target argument, if any, with sequential and keyword arguments taken from the args and kwargs arguments, respectively.

`b2sdk.transferer.parallel.gen_parts (cloud_range, local_range, part_count)`
Generates a sequence of PartToDownload to download a large file as a collection of parts.

4.2.8 `b2sdk.transferer.range`

class `b2sdk.transferer.range.Range (start, end)`
Bases: `object`
HTTP ranges use an *inclusive* index at the end.
__init__ (*start, end*)
Initialize self. See help(type(self)) for accurate signature.
as_tuple ()
classmethod from_header (*raw_range_header*)
factory method which returns an object constructed from Range http header
raw_range_header example: 'bytes 0-11'
size ()
subrange (*sub_start, sub_end*)
Returns a range that is part of this range. :param sub_start: Index relative to the start of this range. :param sub_end: (Inclusive!) index relative to the start of this range. :return: A new Range

4.2.9 `b2sdk.transferer.simple`

class `b2sdk.transferer.simple.SimpleDownloader (*args, **kwargs)`
Bases: `b2sdk.transferer.abstract.AbstractDownloader`
__init__ (**args, **kwargs*)
Initialize self. See help(type(self)) for accurate signature.
download (*file, response, metadata, session*)
@returns (bytes_read, actual_sha1)
is_suitable (*metadata, progress_listener*)
analyzes metadata (possibly against options passed earlier to constructor to find out whether the given download request should be handled by this downloader)

4.2.10 `b2sdk.transferer.simple`

class `b2sdk.transferer.transferer.Transferer (session, account_info)`
Bases: `object`
Handles complex actions around downloads and uploads to free raw_api from that responsibility
DEFAULT_MAX_STREAMS = 8
DEFAULT_MIN_PART_SIZE = 104857600
MAX_CHUNK_SIZE = 1048576
MIN_CHUNK_SIZE = 8192
__init__ (*session, account_info*)
Parameters

- **max_streams** – limit on a number of streams to use when downloading in multiple parts
- **min_part_size** – the smallest part size for which a stream will be run when downloading in multiple parts

download_file_from_url (*url, download_dest, progress_listener=None, range_=None*)

Parameters

- **url** – url from which the file should be downloaded
- **download_dest** – where to put the file when it is downloaded
- **progress_listener** – where to notify about progress downloading
- **range** – 2-element tuple containing data of http Range header

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

b

- b2sdk.account_info.abstract, [15](#)
- b2sdk.account_info.exception, [18](#)
- b2sdk.account_info.sqlite_account_info,
[19](#)
- b2sdk.account_info.upload_url_pool, [21](#)
- b2sdk.api, [25](#)
- b2sdk.b2http, [60](#)
- b2sdk.bucket, [29](#)
- b2sdk.cache, [35](#)
- b2sdk.download_dest, [22](#)
- b2sdk.exception, [39](#)
- b2sdk.file_version, [44](#)
- b2sdk.progress, [36](#)
- b2sdk.raw_api, [57](#)
- b2sdk.session, [57](#)
- b2sdk.sync.action, [45](#)
- b2sdk.sync.exception, [48](#)
- b2sdk.sync.file, [49](#)
- b2sdk.sync.folder, [50](#)
- b2sdk.sync.folder_parser, [49](#)
- b2sdk.sync.policy, [52](#)
- b2sdk.sync.policy_manager, [51](#)
- b2sdk.sync.scan_policies, [54](#)
- b2sdk.sync.sync, [55](#)
- b2sdk.transferer.abstract, [65](#)
- b2sdk.transferer.file_metadata, [65](#)
- b2sdk.transferer.parallel, [66](#)
- b2sdk.transferer.range, [68](#)
- b2sdk.transferer.simple, [68](#)
- b2sdk.transferer.transferer, [68](#)
- b2sdk.upload_source, [24](#)
- b2sdk.utils, [63](#)

Symbols

<code>__init__()</code> (<i>b2sdk.account_info.exception.CorruptAccountInfo</i> method), 18	<code>__init__()</code> (<i>b2sdk.exception.MaxFileSizeExceeded</i> method), 42
<code>__init__()</code> (<i>b2sdk.account_info.exception.MissingAccountData</i> method), 18	<code>__init__()</code> (<i>b2sdk.exception.MaxRetriesExceeded</i> method), 42
<code>__init__()</code> (<i>b2sdk.account_info.sqlite_account_info.SQLiteAccountInfo</i> method), 19	<code>__init__()</code> (<i>b2sdk.exception.PartSha1Mismatch</i> method), 42
<code>__init__()</code> (<i>b2sdk.account_info.upload_url_pool.UploadUrlPool</i> method), 21	<code>__init__()</code> (<i>b2sdk.exception.RestrictedBucket</i> method), 43
<code>__init__()</code> (<i>b2sdk.account_info.upload_url_pool.UrlPoolAccountInfo</i> method), 21	<code>__init__()</code> (<i>b2sdk.exception.TruncatedOutput</i> method), 43
<code>__init__()</code> (<i>b2sdk.api.B2Api</i> method), 25	<code>__init__()</code> (<i>b2sdk.exception.Unauthorized</i> method), 43
<code>__init__()</code> (<i>b2sdk.b2http.B2Http</i> method), 60	<code>__init__()</code> (<i>b2sdk.file_version.FileIdAndName</i> method), 44
<code>__init__()</code> (<i>b2sdk.b2http.ResponseContextManager</i> method), 63	<code>__init__()</code> (<i>b2sdk.file_version.FileVersionInfo</i> method), 44
<code>__init__()</code> (<i>b2sdk.bucket.Bucket</i> method), 29	<code>__init__()</code> (<i>b2sdk.progress.AbstractProgressListener</i> method), 36
<code>__init__()</code> (<i>b2sdk.bucket.LargeFileUploadState</i> method), 34	<code>__init__()</code> (<i>b2sdk.progress.AbstractStreamWithProgress</i> method), 36
<code>__init__()</code> (<i>b2sdk.bucket.PartProgressReporter</i> method), 34	<code>__init__()</code> (<i>b2sdk.progress.ProgressListenerForTest</i> method), 37
<code>__init__()</code> (<i>b2sdk.cache.AuthInfoCache</i> method), 35	<code>__init__()</code> (<i>b2sdk.progress.RangeOfInputStream</i> method), 37
<code>__init__()</code> (<i>b2sdk.cache.InMemoryCache</i> method), 35	<code>__init__()</code> (<i>b2sdk.progress.SimpleProgressListener</i> method), 38
<code>__init__()</code> (<i>b2sdk.download_dest.DownloadDestBytes</i> method), 22	<code>__init__()</code> (<i>b2sdk.progress.StreamWithHash</i> method), 38
<code>__init__()</code> (<i>b2sdk.download_dest.DownloadDestLocalFile</i> method), 23	<code>__init__()</code> (<i>b2sdk.progress.TqdmProgressListener</i> method), 39
<code>__init__()</code> (<i>b2sdk.download_dest.DownloadDestProgressWrapper</i> method), 24	<code>__init__()</code> (<i>b2sdk.raw_api.B2RawApi</i> method), 58
<code>__init__()</code> (<i>b2sdk.download_dest.PreSeekedDownloadDest</i> method), 24	<code>__init__()</code> (<i>b2sdk.session.B2Session</i> method), 57
<code>__init__()</code> (<i>b2sdk.exception.B2Error</i> method), 39	<code>__init__()</code> (<i>b2sdk.sync.action.B2DeleteAction</i> method), 45
<code>__init__()</code> (<i>b2sdk.exception.ChecksumMismatch</i> method), 40	<code>__init__()</code> (<i>b2sdk.sync.action.B2DownloadAction</i> method), 46
<code>__init__()</code> (<i>b2sdk.exception.ClockSkew</i> method), 41	<code>__init__()</code> (<i>b2sdk.sync.action.B2HideAction</i> method), 47
<code>__init__()</code> (<i>b2sdk.exception.CommandError</i> method), 41	<code>__init__()</code> (<i>b2sdk.sync.action.B2UploadAction</i> method), 47
<code>__init__()</code> (<i>b2sdk.exception.DestFileNewer</i> method), 41	<code>__init__()</code> (<i>b2sdk.sync.action.LocalDeleteAction</i> method), 47
<code>__init__()</code> (<i>b2sdk.exception.InvalidAuthToken</i> method), 41	
<code>__init__()</code> (<i>b2sdk.exception.InvalidRange</i> method), 42	

method), 48
 __init__() (b2sdk.sync.exception.EnvironmentEncodingError method), 48
 __init__() (b2sdk.sync.file.File method), 49
 __init__() (b2sdk.sync.file.FileVersion method), 49
 __init__() (b2sdk.sync.folder.B2Folder method), 50
 __init__() (b2sdk.sync.folder.LocalFolder method), 51
 __init__() (b2sdk.sync.policy.AbstractFileSyncPolicy method), 52
 __init__() (b2sdk.sync.policy_manager.SyncPolicyManager method), 51
 __init__() (b2sdk.sync.scan_policies.RegexSet method), 54
 __init__() (b2sdk.sync.scan_policies.ScanPoliciesManager method), 54
 __init__() (b2sdk.transferer.abstract.AbstractDownloader method), 65
 __init__() (b2sdk.transferer.file_metadata.FileMetadata method), 65
 __init__() (b2sdk.transferer.parallel.AbstractDownloaderThread method), 66
 __init__() (b2sdk.transferer.parallel.FirstPartDownloaderThread method), 66
 __init__() (b2sdk.transferer.parallel.NonHashingDownloaderThread method), 66
 __init__() (b2sdk.transferer.parallel.ParallelDownloader method), 67
 __init__() (b2sdk.transferer.parallel.PartToDownload method), 67
 __init__() (b2sdk.transferer.parallel.WriterThread method), 67
 __init__() (b2sdk.transferer.range.Range method), 68
 __init__() (b2sdk.transferer.simple.SimpleDownloader method), 68
 __init__() (b2sdk.transferer.transferer.Transferer method), 68
 __init__() (b2sdk.upload_source.UploadSourceBytes method), 25
 __init__() (b2sdk.upload_source.UploadSourceLocalFile method), 25
 __init__() (b2sdk.utils.BytesIoContextManager method), 63

AbstractDownloaderThread (class in b2sdk.transferer.parallel), 66
 AbstractFileSyncPolicy (class in b2sdk.sync.policy), 52
 AbstractFolder (class in b2sdk.sync.folder), 50
 AbstractProgressListener (class in b2sdk.progress), 36
 AbstractRawApi (class in b2sdk.raw_api), 57
 AbstractStreamWithProgress (class in b2sdk.progress), 36
 AbstractUploadSource (class in b2sdk.upload_source), 24
 AccountInfoError, 18
 add_callback() (b2sdk.b2http.B2Http method), 61
 get_capabilities() (b2sdk.account_info.abstract.AbstractAccountInfo class method), 15
 all_files() (b2sdk.sync.folder.AbstractFolder method), 50
 all_files() (b2sdk.sync.folder.B2Folder method), 50
 all_files() (b2sdk.sync.folder.LocalFolder method), 51
 is_valid() (b2sdk.account_info.abstract.AbstractAccountInfo class method), 15
 AlreadyFailed, 39
 as_dict() (b2sdk.bucket.Bucket method), 29
 as_dict() (b2sdk.file_version.FileIdAndName method), 44
 as_dict() (b2sdk.file_version.FileVersionInfo method), 44
 as_info_dict() (b2sdk.transferer.file_metadata.FileMetadata method), 65
 as_tuple() (b2sdk.transferer.range.Range method), 68
 AuthInfoCache (class in b2sdk.cache), 35
 authorize_account() (b2sdk.api.B2Api method), 26
 authorize_account() (b2sdk.raw_api.B2RawApi method), 58
 authorize_automatically() (b2sdk.api.B2Api method), 26

A

AbstractAccountInfo (class in b2sdk.account_info.abstract), 15
 AbstractAction (class in b2sdk.sync.action), 45
 AbstractCache (class in b2sdk.cache), 35
 AbstractDownloadDestination (class in b2sdk.download_dest), 22
 AbstractDownloader (class in b2sdk.transferer.abstract), 65

B

b2_url_decode() (in module b2sdk.utils), 63
 b2_url_encode() (in module b2sdk.utils), 63
 B2Api (class in b2sdk.api), 25
 B2ConnectionError, 39
 B2DeleteAction (class in b2sdk.sync.action), 45
 B2DownloadAction (class in b2sdk.sync.action), 46
 B2Error, 39
 B2Folder (class in b2sdk.sync.folder), 50
 B2HideAction (class in b2sdk.sync.action), 47
 B2Http (class in b2sdk.b2http), 60

B2HttpCallbackException, 40
 B2HttpCallbackPostRequestException, 40
 B2HttpCallbackPreRequestException, 40
 B2RawApi (class in *b2sdk.raw_api*), 58
 B2RequestTimeout, 40
 b2sdk.account_info.abstract (module), 15
 b2sdk.account_info.exception (module), 18
 b2sdk.account_info.sqlite_account_info (module), 19
 b2sdk.account_info.upload_url_pool (module), 21
 b2sdk.api (module), 25
 b2sdk.b2http (module), 60
 b2sdk.bucket (module), 29
 b2sdk.cache (module), 35
 b2sdk.download_dest (module), 22
 b2sdk.exception (module), 39
 b2sdk.file_version (module), 44
 b2sdk.progress (module), 36
 b2sdk.raw_api (module), 57
 b2sdk.session (module), 57
 b2sdk.sync.action (module), 45
 b2sdk.sync.exception (module), 48
 b2sdk.sync.file (module), 49
 b2sdk.sync.folder (module), 50
 b2sdk.sync.folder_parser (module), 49
 b2sdk.sync.policy (module), 52
 b2sdk.sync.policy_manager (module), 51
 b2sdk.sync.scan_policies (module), 54
 b2sdk.sync.sync (module), 55
 b2sdk.transferer.abstract (module), 65
 b2sdk.transferer.file_metadata (module), 65
 b2sdk.transferer.parallel (module), 66
 b2sdk.transferer.range (module), 68
 b2sdk.transferer.simple (module), 68
 b2sdk.transferer.transferer (module), 68
 b2sdk.upload_source (module), 24
 b2sdk.utils (module), 63
 B2Session (class in *b2sdk.session*), 57
 B2SimpleError, 40
 B2TraceMeta (class in *b2sdk.utils*), 63
 B2TraceMetaAbstract (class in *b2sdk.utils*), 63
 B2UploadAction (class in *b2sdk.sync.action*), 47
 BadDateFormat, 40
 BadFileInfo, 40
 BadJson, 40
 BadUploadUrl, 40
 BrokenPipe, 40
 Bucket (class in *b2sdk.bucket*), 29
 BucketFactory (class in *b2sdk.bucket*), 33
 BucketNotAllowed, 40
 bytes_completed() (b2sdk.bucket.PartProgressReporter method), 34
 bytes_completed() (b2sdk.progress.AbstractProgressListener method), 36
 bytes_completed() (b2sdk.progress.DoNothingProgressListener method), 37
 bytes_completed() (b2sdk.progress.ProgressListenerForTest method), 37
 bytes_completed() (b2sdk.progress.SimpleProgressListener method), 38
 bytes_completed() (b2sdk.progress.TqdmProgressListener method), 39
 BytesIOContextManager (class in *b2sdk.utils*), 63

C

camelcase_to_underscore() (in module *b2sdk.utils*), 63
 cancel_large_file() (b2sdk.api.B2Api method), 26
 cancel_large_file() (b2sdk.bucket.Bucket method), 29
 cancel_large_file() (b2sdk.raw_api.AbstractRawApi method), 57
 cancel_large_file() (b2sdk.raw_api.B2RawApi method), 58
 CapabilityNotAllowed, 40
 capture_bytes_context() (b2sdk.download_dest.DownloadDestBytes method), 23
 check_b2_filename() (b2sdk.raw_api.B2RawApi method), 58
 check_bucket_restrictions() (b2sdk.api.B2Api method), 26
 ChecksumMismatch, 40
 choose_part_ranges() (in module *b2sdk.utils*), 64
 clear() (b2sdk.account_info.abstract.AbstractAccountInfo method), 15
 clear() (b2sdk.account_info.sqlite_account_info.SqliteAccountInfo method), 19
 clear() (b2sdk.account_info.upload_url_pool.UrlPoolAccountInfo method), 21
 clear() (b2sdk.cache.AbstractCache method), 35
 clear_bucket_upload_data() (b2sdk.account_info.abstract.AbstractAccountInfo method), 15
 clear_bucket_upload_data() (b2sdk.account_info.upload_url_pool.UrlPoolAccountInfo method), 21

`clear_for_key()` (`b2sdk.account_info.upload_url_pool.UrlPoolStreams` method), 21
`clear_large_file_upload_urls()` (`b2sdk.account_info.abstract.AbstractAccountInfo` method), 16
`clear_large_file_upload_urls()` (`b2sdk.account_info.upload_url_pool.UrlPoolAccountInfo` method), 21
`ClockSkew`, 40
`ClockSkewHook` (class in `b2sdk.b2http`), 62
`close()` (`b2sdk.bucket.PartProgressReporter` method), 34
`close()` (`b2sdk.progress.AbstractProgressListener` method), 36
`close()` (`b2sdk.progress.DoNothingProgressListener` method), 37
`close()` (`b2sdk.progress.ProgressListenerForTest` method), 37
`close()` (`b2sdk.progress.SimpleProgressListener` method), 38
`close()` (`b2sdk.progress.TqdmProgressListener` method), 39
`CommandError`, 41
`Conflict`, 41
`ConnectionReset`, 41
`content_length` (`b2sdk.transferer.file_metadata.FileMetadata` attribute), 65
`content_shal` (`b2sdk.transferer.file_metadata.FileMetadata` attribute), 65
`content_type` (`b2sdk.transferer.file_metadata.FileMetadata` attribute), 65
`convert_dir_regex_to_dir_prefix_regex()` (in module `b2sdk.sync.scan_policies`), 55
`COPY` (`b2sdk.raw_api.MetadataDirectiveMode` attribute), 60
`copy_file()` (`b2sdk.bucket.Bucket` method), 29
`copy_file()` (`b2sdk.raw_api.AbstractRawApi` method), 57
`copy_file()` (`b2sdk.raw_api.B2RawApi` method), 58
`CorruptAccountInfo`, 18
`count_files()` (in module `b2sdk.sync.sync`), 55
`create_bucket()` (`b2sdk.api.B2Api` method), 26
`create_bucket()` (`b2sdk.raw_api.B2RawApi` method), 58
`create_key()` (`b2sdk.api.B2Api` method), 26
`create_key()` (`b2sdk.raw_api.B2RawApi` method), 58
D
`DEFAULT_ALLOWED` (`b2sdk.account_info.abstract.AbstractAccountInfo` attribute), 15
`DEFAULT_CONTENT_TYPE` (`b2sdk.bucket.Bucket` attribute), 29
`DEFAULT_UPLOAD_URL` (`b2sdk.transferer.transferer.Transferer` attribute), 68
`DEFAULT_MIN_PART_SIZE` (`b2sdk.transferer.transferer.Transferer` attribute), 68
`delete_bucket()` (`b2sdk.api.B2Api` method), 26
`delete_bucket()` (`b2sdk.raw_api.AbstractRawApi` method), 57
`delete_bucket()` (`b2sdk.raw_api.B2RawApi` method), 58
`delete_file_version()` (`b2sdk.api.B2Api` method), 27
`delete_file_version()` (`b2sdk.bucket.Bucket` method), 30
`delete_file_version()` (`b2sdk.raw_api.AbstractRawApi` method), 57
`delete_file_version()` (`b2sdk.raw_api.B2RawApi` method), 58
`delete_key()` (`b2sdk.api.B2Api` method), 27
`delete_key()` (`b2sdk.raw_api.B2RawApi` method), 58
`DestFileNewer`, 41
`DESTINATION_PREFIX` (`b2sdk.sync.policy.AbstractFileSyncPolicy` attribute), 52
`DESTINATION_PREFIX` (`b2sdk.sync.policy.DownPolicy` attribute), 53
`DESTINATION_PREFIX` (`b2sdk.sync.policy.UpPolicy` attribute), 53
`do_action()` (`b2sdk.sync.action.AbstractAction` method), 45
`do_action()` (`b2sdk.sync.action.B2DeleteAction` method), 46
`do_action()` (`b2sdk.sync.action.B2DownloadAction` method), 46
`do_action()` (`b2sdk.sync.action.B2HideAction` method), 47
`do_action()` (`b2sdk.sync.action.B2UploadAction` method), 47
`do_action()` (`b2sdk.sync.action.LocalDeleteAction` method), 48
`do_report()` (`b2sdk.sync.action.AbstractAction` method), 45
`do_report()` (`b2sdk.sync.action.B2DeleteAction` method), 46
`do_report()` (`b2sdk.sync.action.B2DownloadAction` method), 46
`do_report()` (`b2sdk.sync.action.B2HideAction` method), 47
`do_report()` (`b2sdk.sync.action.B2UploadAction` method), 48

do_report() (*b2sdk.sync.action.LocalDeleteAction* *method*), 48

DoNothingProgressListener (*class* *in* *b2sdk.progress*), 37

DownAndDeletePolicy (*class* *in* *b2sdk.sync.policy*), 53

DownAndKeepDaysPolicy (*class* *in* *b2sdk.sync.policy*), 53

download() (*b2sdk.transferer.abstract.AbstractDownloader* *method*), 65

download() (*b2sdk.transferer.parallel.ParallelDownloader* *method*), 67

download() (*b2sdk.transferer.simple.SimpleDownloader* *method*), 68

download_file_by_id() (*b2sdk.api.B2Api* *method*), 27

download_file_by_id() (*b2sdk.bucket.Bucket* *method*), 30

download_file_by_name() (*b2sdk.bucket.Bucket* *method*), 30

download_file_from_url() (*b2sdk.raw_api.B2RawApi* *method*), 58

download_file_from_url() (*b2sdk.transferer.transferer.Transferer* *method*), 69

DownloadDestBytes (*class* *in* *b2sdk.download_dest*), 22

DownloadDestLocalFile (*class* *in* *b2sdk.download_dest*), 23

DownloadDestProgressWrapper (*class* *in* *b2sdk.download_dest*), 23

DownPolicy (*class* *in* *b2sdk.sync.policy*), 53

DummyCache (*class* *in* *b2sdk.cache*), 35

DuplicateBucketName, 41

E

ensure_non_empty() (*b2sdk.sync.folder.LocalFolder* *method*), 51

ensure_present() (*b2sdk.sync.folder.LocalFolder* *method*), 51

EnvironmentEncodingError, 48

F

File (*class* *in* *b2sdk.sync.file*), 49

file_id (*b2sdk.transferer.file_metadata.FileMetadata* *attribute*), 65

file_info (*b2sdk.transferer.file_metadata.FileMetadata* *attribute*), 65

file_name (*b2sdk.transferer.file_metadata.FileMetadata* *attribute*), 66

FileAlreadyHidden, 41

FileIdAndName (*class* *in* *b2sdk.file_version*), 44

FileMetadata (*class* *in* *b2sdk.transferer.file_metadata*), 65

FileNameNotAllowed, 41

FileNotPresent, 41

files_are_different() (*b2sdk.sync.policy.AbstractFileSyncPolicy* *class method*), 52

FileVersion (*class* *in* *b2sdk.sync.file*), 49

FileVersionInfo (*class* *in* *b2sdk.file_version*), 44

FileVersionInfoFactory (*class* *in* *b2sdk.file_version*), 44

FINISH_HASHING_BUFFER_SIZE (*b2sdk.transferer.parallel.ParallelDownloader* *attribute*), 67

finish_large_file() (*b2sdk.raw_api.AbstractRawApi* *method*), 57

finish_large_file() (*b2sdk.raw_api.B2RawApi* *method*), 59

FirstPartDownloaderThread (*class* *in* *b2sdk.transferer.parallel*), 66

fix_windows_path_limit() (*in* *module* *b2sdk.utils*), 64

flush() (*b2sdk.progress.AbstractStreamWithProgress* *method*), 36

folder_type() (*b2sdk.sync.folder.AbstractFolder* *method*), 50

folder_type() (*b2sdk.sync.folder.B2Folder* *method*), 50

folder_type() (*b2sdk.sync.folder.LocalFolder* *method*), 51

format_and_scale_fraction() (*in* *module* *b2sdk.utils*), 64

format_and_scale_number() (*in* *module* *b2sdk.utils*), 64

format_folder_ls_entry() (*b2sdk.file_version.FileVersionInfo* *class method*), 44

format_ls_entry() (*b2sdk.file_version.FileVersionInfo* *method*), 44

from_api_bucket_dict() (*b2sdk.bucket.BucketFactory* *class method*), 33

from_api_response() (*b2sdk.bucket.BucketFactory* *class method*), 33

from_api_response() (*b2sdk.file_version.FileVersionInfoFactory* *class method*), 44

from_cancel_large_file_response() (*b2sdk.file_version.FileVersionInfoFactory* *class method*), 45

from_header() (*b2sdk.transferer.range.Range* *class method*), 68

from_response() (*b2sdk.transferer.file_metadata.FileMetadata*

class method), 66

G

gen_parts() (in module *b2sdk.transferer.parallel*), 67

get_account_auth_token() (*b2sdk.account_info.abstract.AbstractAccountInfo method*), 16

get_account_auth_token() (*b2sdk.account_info.sqlite_account_info.SqliteAccountInfo method*), 19

get_account_id() (*b2sdk.account_info.abstract.AbstractAccountInfo method*), 16

get_account_id() (*b2sdk.account_info.sqlite_account_info.SqliteAccountInfo method*), 19

get_account_id() (*b2sdk.api.B2Api method*), 27

get_account_id_or_app_key_id() (*b2sdk.account_info.abstract.AbstractAccountInfo method*), 16

get_all_actions() (*b2sdk.sync.policy.AbstractFileSyncPolicy method*), 53

get_allowed() (*b2sdk.account_info.abstract.AbstractAccountInfo method*), 16

get_allowed() (*b2sdk.account_info.sqlite_account_info.SqliteAccountInfo method*), 19

get_api_url() (*b2sdk.account_info.abstract.AbstractAccountInfo method*), 16

get_api_url() (*b2sdk.account_info.sqlite_account_info.SqliteAccountInfo method*), 19

get_application_key() (*b2sdk.account_info.abstract.AbstractAccountInfo method*), 16

get_application_key() (*b2sdk.account_info.sqlite_account_info.SqliteAccountInfo method*), 19

get_application_key_id() (*b2sdk.account_info.abstract.AbstractAccountInfo method*), 16

get_application_key_id() (*b2sdk.account_info.sqlite_account_info.SqliteAccountInfo method*), 19

get_bucket_by_id() (*b2sdk.api.B2Api method*), 27

get_bucket_by_name() (*b2sdk.api.B2Api method*), 27

get_bucket_id_or_none_from_bucket_name() (*b2sdk.account_info.abstract.AbstractAccountInfo method*), 16

get_bucket_id_or_none_from_bucket_name() (*b2sdk.account_info.sqlite_account_info.SqliteAccountInfo method*), 20

get_bucket_id_or_none_from_bucket_name() (*b2sdk.cache.AbstractCache method*), 35

get_bucket_id_or_none_from_bucket_name() (*b2sdk.cache.AuthInfoCache method*), 35

get_bucket_id_or_none_from_bucket_name() (*b2sdk.cache.DummyCache method*), 35

get_bucket_id_or_none_from_bucket_name() (*b2sdk.cache.InMemoryCache method*), 35

get_bucket_name_or_none_from_allowed() (*b2sdk.cache.AbstractCache method*), 35

get_bucket_name_or_none_from_allowed() (*b2sdk.cache.AuthInfoCache method*), 35

get_bucket_name_or_none_from_allowed() (*b2sdk.cache.DummyCache method*), 35

get_bucket_name_or_none_from_allowed() (*b2sdk.cache.InMemoryCache method*), 35

get_bytes() (*b2sdk.sync.action.AbstractAction method*), 45

get_bytes() (*b2sdk.sync.action.B2DeleteAction method*), 46

get_bytes() (*b2sdk.sync.action.B2DownloadAction method*), 47

get_bytes() (*b2sdk.sync.action.B2HideAction method*), 47

get_bytes() (*b2sdk.sync.action.B2UploadAction method*), 48

get_bytes() (*b2sdk.sync.action.LocalDeleteAction method*), 48

get_bytes_written() (*b2sdk.download_dest.DownloadDestBytes method*), 23

get_calls() (*b2sdk.progress.ProgressListenerForTest method*), 37

get_content() (*b2sdk.b2http.B2Http method*), 61

get_content_length() (*b2sdk.upload_source.AbstractUploadSource method*), 24

get_content_length() (*b2sdk.upload_source.UploadSourceBytes method*), 25

get_content_length() (*b2sdk.upload_source.UploadSourceLocalFile method*), 25

get_content_sha1() (*b2sdk.upload_source.AbstractUploadSource method*), 24

get_content_sha1() (*b2sdk.upload_source.UploadSourceBytes method*), 25

get_content_sha1() (*b2sdk.upload_source.UploadSourceLocalFile method*), 25

get_download_authorization() (*b2sdk.bucket.Bucket method*), 30

get_download_authorization() (*b2sdk.raw_api.B2RawApi method*), 59

get_download_url() (*b2sdk.account_info.abstract.AbstractAccountInfo method*), 16

method), 16
 get_download_url() (b2sdk.account_info.sqlite_account_info.SqliteAccountInfo *method*), 57
method), 20
 get_download_url() (b2sdk.bucket.Bucket *method*), 30
 get_download_url_by_id() (b2sdk.raw_api.AbstractRawApi *method*), 57
 get_download_url_by_name() (b2sdk.raw_api.AbstractRawApi *method*), 57
 get_download_url_for_file_name() (b2sdk.api.B2Api *method*), 27
 get_download_url_for_fileid() (b2sdk.api.B2Api *method*), 28
 get_error_message() (b2sdk.bucket.LargeFileUploadState *method*), 34
 get_file_info() (b2sdk.api.B2Api *method*), 28
 get_file_info() (b2sdk.raw_api.B2RawApi *method*), 59
 get_id() (b2sdk.bucket.Bucket *method*), 30
 get_minimum_part_size() (b2sdk.account_info.abstract.AbstractAccountInfo *method*), 16
 get_minimum_part_size() (b2sdk.account_info.sqlite_account_info.SqliteAccountInfo *method*), 20
 get_policy() (b2sdk.sync.policy_manager.SyncPolicyManager *method*), 51
 get_policy_class() (b2sdk.sync.policy_manager.SyncPolicyManager *method*), 52
 get_realm() (b2sdk.account_info.abstract.AbstractAccountInfo *method*), 16
 get_realm() (b2sdk.account_info.sqlite_account_info.SqliteAccountInfo *method*), 20
 get_thread_pool() (b2sdk.api.B2Api *method*), 28
 get_upload_part_url() (b2sdk.raw_api.AbstractRawApi *method*), 57
 get_upload_part_url() (b2sdk.raw_api.B2RawApi *method*), 59
 get_upload_url() (b2sdk.raw_api.B2RawApi *method*), 59

H

has_error() (b2sdk.bucket.LargeFileUploadState *method*), 34
 hash_size() (b2sdk.progress.StreamWithHash *method*), 38
 hex_shal_of_bytes() (in module b2sdk.utils), 64
 hex_shal_of_stream() (in module b2sdk.utils), 64

hide_file() (b2sdk.bucket.Bucket *method*), 30
 hide_file() (b2sdk.raw_api.AbstractRawApi *method*), 59
 hide_file() (b2sdk.raw_api.B2RawApi *method*), 59
 HttpCallback (class in b2sdk.b2http), 62

I

InMemoryCache (class in b2sdk.cache), 35
 interpret_b2_error() (in module b2sdk.exception), 44
 interruptible_get_result() (in module b2sdk.utils), 65
 InvalidAuthToken, 41
 InvalidMetadataDirective, 42
 InvalidRange, 42
 InvalidUploadSource, 42
 is_file_readable() (in module b2sdk.utils), 65
 is_suitable() (b2sdk.transferer.abstract.AbstractDownloader *method*), 65
 is_suitable() (b2sdk.transferer.parallel.ParallelDownloader *method*), 67
 is_suitable() (b2sdk.transferer.simple.SimpleDownloader *method*), 68

J

join_b2_path() (in module b2sdk.sync.folder), 51

LargeFileUploadState (class in b2sdk.bucket), 34
 latest_version() (b2sdk.sync.file.File *method*), 49
 list_buckets() (b2sdk.api.B2Api *method*), 28
 list_buckets() (b2sdk.raw_api.B2RawApi *method*), 59
 list_file_names() (b2sdk.bucket.Bucket *method*), 31
 list_file_names() (b2sdk.raw_api.B2RawApi *method*), 59
 list_file_versions() (b2sdk.bucket.Bucket *method*), 31
 list_file_versions() (b2sdk.raw_api.B2RawApi *method*), 59
 list_keys() (b2sdk.api.B2Api *method*), 28
 list_keys() (b2sdk.raw_api.B2RawApi *method*), 59
 list_parts() (b2sdk.api.B2Api *method*), 28
 list_parts() (b2sdk.bucket.Bucket *method*), 31
 list_parts() (b2sdk.raw_api.AbstractRawApi *method*), 58
 list_parts() (b2sdk.raw_api.B2RawApi *method*), 59
 list_unfinished_large_files() (b2sdk.bucket.Bucket *method*), 31
 list_unfinished_large_files() (b2sdk.raw_api.AbstractRawApi *method*), 58

list_unfinished_large_files()
 (*b2sdk.raw_api.B2RawApi* method), 59
 LocalDeleteAction (class in *b2sdk.sync.action*), 48
 LocalFolder (class in *b2sdk.sync.folder*), 51
 ls() (*b2sdk.bucket.Bucket* method), 31
 LS_ENTRY_TEMPLATE
 (*b2sdk.file_version.FileVersionInfo* attribute),
 44

M

make_b2_delete_actions() (in module
 b2sdk.sync.policy), 53
 make_b2_delete_note() (in module
 b2sdk.sync.policy), 53
 make_b2_keep_days_actions() (in module
 b2sdk.sync.policy), 54
 make_file_context()
 (*b2sdk.download_dest.AbstractDownloadDestination*
 method), 22
 make_file_context()
 (*b2sdk.download_dest.DownloadDestBytes*
 method), 23
 make_file_context()
 (*b2sdk.download_dest.DownloadDestLocalFile*
 method), 23
 make_file_context()
 (*b2sdk.download_dest.DownloadDestProgressWrapper*
 method), 24
 make_file_sync_actions() (in module
 b2sdk.sync.sync), 55
 make_folder_sync_actions() (in module
 b2sdk.sync.sync), 56
 make_full_path() (*b2sdk.sync.folder.AbstractFolder*
 method), 50
 make_full_path() (*b2sdk.sync.folder.B2Folder*
 method), 50
 make_full_path() (*b2sdk.sync.folder.LocalFolder*
 method), 51
 make_progress_listener() (in module
 b2sdk.progress), 39
 matches() (*b2sdk.sync.scan_policies.RegexSet*
 method), 54
 MAX_CHUNK_SIZE (*b2sdk.transferer.transferer.Transferer*
 attribute), 68
 MAX_LARGE_FILE_SIZE (*b2sdk.bucket.Bucket* at-
 tribute), 29
 MAX_UPLOAD_ATTEMPTS (*b2sdk.bucket.Bucket* at-
 tribute), 29
 MaxFileSizeExceeded, 42
 MaxRetriesExceeded, 42
 MetadataDirectiveMode (class in *b2sdk.raw_api*),
 60
 MIN_CHUNK_SIZE (*b2sdk.transferer.transferer.Transferer*
 attribute), 68

MissingAccountData, 18
 MissingPart, 42
 MODE (*b2sdk.download_dest.DownloadDestLocalFile* at-
 tribute), 23
 MODE (*b2sdk.download_dest.PreSeekedDownloadDest*
 attribute), 24

N

next_or_none() (in module *b2sdk.sync.sync*), 56
 NonExistentBucket, 42
 NonHashingDownloaderThread (class in
 b2sdk.transferer.parallel), 66
 NotAllowedByAppKeyError, 42

O

open() (*b2sdk.upload_source.AbstractUploadSource*
 method), 24
 open() (*b2sdk.upload_source.UploadSourceBytes*
 method), 25
 open() (*b2sdk.upload_source.UploadSourceLocalFile*
 method), 25

P

ParallelDownloader (class in
 b2sdk.transferer.parallel), 66
 parse_sync_folder() (in module
 b2sdk.sync.folder_parser), 49
 PartProgressReporter (class in *b2sdk.bucket*), 34
 PartShalMismatch, 42
 PartToDownload (class in *b2sdk.transferer.parallel*),
 67
 post_content_return_json()
 (*b2sdk.b2http.B2Http* method), 61
 post_json_return_json() (*b2sdk.b2http.B2Http*
 method), 61
 post_request() (*b2sdk.b2http.ClockSkewHook*
 method), 62
 post_request() (*b2sdk.b2http.HttpCallback*
 method), 62
 pre_request() (*b2sdk.b2http.HttpCallback* method),
 62
 prefix (*b2sdk.exception.BadDateFormat* attribute), 40
 prefix (*b2sdk.exception.BadJson* attribute), 40
 prefix (*b2sdk.exception.DuplicateBucketName* at-
 tribute), 41
 prefix (*b2sdk.exception.MissingPart* attribute), 42
 prefix (*b2sdk.exception.NonExistentBucket* attribute),
 42
 prefix() (*b2sdk.exception.B2Error* property), 40
 PreSeekedDownloadDest (class in
 b2sdk.download_dest), 24
 ProgressListenerForTest (class in
 b2sdk.progress), 37

put () (*b2sdk.account_info.upload_url_pool.UploadUrlPool* method), 21
 put_bucket_upload_url () (*b2sdk.account_info.abstract.AbstractAccountInfo* method), 17
 put_bucket_upload_url () (*b2sdk.account_info.upload_url_pool.UrlPoolAccountInfo* method), 21
 put_large_file_upload_url () (*b2sdk.account_info.abstract.AbstractAccountInfo* method), 17
 put_large_file_upload_url () (*b2sdk.account_info.upload_url_pool.UrlPoolAccountInfo* method), 22
 Range (*class in b2sdk.transferer.range*), 68
 RangeOfInputStream (*class in b2sdk.progress*), 37
 read () (*b2sdk.progress.AbstractStreamWithProgress* method), 36
 read () (*b2sdk.progress.RangeOfInputStream* method), 37
 read () (*b2sdk.progress.ReadingStreamWithProgress* method), 38
 read () (*b2sdk.progress.StreamWithHash* method), 38
 ReadingStreamWithProgress (*class in b2sdk.progress*), 38
 REALM_URLS (*b2sdk.account_info.abstract.AbstractAccountInfo* attribute), 15
 refresh_entire_bucket_name_cache () (*b2sdk.account_info.abstract.AbstractAccountInfo* method), 17
 refresh_entire_bucket_name_cache () (*b2sdk.account_info.sqlite_account_info.SqliteAccountInfo* method), 20
 RegexSet (*class in b2sdk.sync.scan_policies*), 54
 remove_bucket_name () (*b2sdk.account_info.abstract.AbstractAccountInfo* method), 17
 remove_bucket_name () (*b2sdk.account_info.sqlite_account_info.SqliteAccountInfo* method), 20
 REPLACE (*b2sdk.raw_api.MetadataDirectiveMode* attribute), 60
 ResponseContextManager (*class in b2sdk.b2http*), 63
 RestrictedBucket, 43
 run () (*b2sdk.sync.action.AbstractAction* method), 45
 run () (*b2sdk.transferer.parallel.AbstractDownloaderThread* method), 66
 run () (*b2sdk.transferer.parallel.FirstPartDownloaderThread* method), 66
 run () (*b2sdk.transferer.parallel.NonHashingDownloaderThread* method), 66
 run () (*b2sdk.transferer.parallel.WriterThread* method), 67
 save_bucket () (*b2sdk.account_info.abstract.AbstractAccountInfo* method), 17
 save_bucket () (*b2sdk.account_info.sqlite_account_info.SqliteAccountInfo* method), 20
 save_bucket () (*b2sdk.cache.AbstractCache* method), 35
 save_bucket () (*b2sdk.cache.AuthInfoCache* method), 35
 save_bucket () (*b2sdk.cache.DummyCache* method), 35
 save_bucket () (*b2sdk.cache.InMemoryCache* method), 35
 ScanPoliciesManager (*class in b2sdk.sync.scan_policies*), 54
 seek () (*b2sdk.progress.AbstractStreamWithProgress* method), 36
 seek () (*b2sdk.progress.RangeOfInputStream* method), 38
 seek () (*b2sdk.progress.StreamWithHash* method), 38
 ServiceError, 43
 set_auth_data () (*b2sdk.account_info.abstract.AbstractAccountInfo* method), 17
 set_auth_data_with_schema_0_for_test () (*b2sdk.account_info.sqlite_account_info.SqliteAccountInfo* method), 20
 set_bucket_name_cache () (*b2sdk.cache.AbstractCache* method), 35
 set_bucket_name_cache () (*b2sdk.cache.AuthInfoCache* method), 35
 set_bucket_name_cache () (*b2sdk.cache.DummyCache* method), 35
 set_bucket_name_cache () (*b2sdk.cache.InMemoryCache* method), 35
 set_error () (*b2sdk.bucket.LargeFileUploadState* method), 34
 set_info () (*b2sdk.bucket.Bucket* method), 31
 set_thread_pool_size () (*b2sdk.api.B2Api* method), 28
 set_total_bytes () (*b2sdk.bucket.PartProgressReporter* method), 34
 set_total_bytes () (*b2sdk.progress.AbstractProgressListener* method), 36
 set_total_bytes () (*b2sdk.progress.DoNothingProgressListener* method), 37
 set_total_bytes () (*b2sdk.progress.ProgressListenerForTest* method), 37

method), 37
 set_total_bytes() (b2sdk.progress.SimpleProgressListener method), 38
 set_total_bytes() (b2sdk.progress.TqdmProgressListener method), 39
 set_type() (b2sdk.bucket.Bucket method), 32
 should_exclude_directory() (b2sdk.sync.scan_policies.ScanPoliciesManager method), 55
 should_exclude_file() (b2sdk.sync.scan_policies.ScanPoliciesManager method), 55
 should_retry_http() (b2sdk.exception.B2Error method), 40
 should_retry_http() (b2sdk.exception.DestFileNewer method), 41
 should_retry_http() (b2sdk.exception.TooManyRequests method), 43
 should_retry_http() (b2sdk.exception.TransientErrorMixin method), 43
 should_retry_upload() (b2sdk.exception.B2Error method), 40
 should_retry_upload() (b2sdk.exception.BrokenPipe method), 40
 should_retry_upload() (b2sdk.exception.ConnectionReset method), 41
 should_retry_upload() (b2sdk.exception.TransientErrorMixin method), 43
 should_retry_upload() (b2sdk.exception.Unauthorized method), 43
 SimpleDownloader (class in b2sdk.transferer.simple), 68
 SimpleProgressListener (class in b2sdk.progress), 38
 size() (b2sdk.transferer.range.Range method), 68
 SOURCE_PREFIX (b2sdk.sync.policy.AbstractFileSyncPolicy attribute), 52
 SOURCE_PREFIX (b2sdk.sync.policy.DownPolicy attribute), 53
 SOURCE_PREFIX (b2sdk.sync.policy.UpPolicy attribute), 53
 SqliteAccountInfo (class in b2sdk.account_info.sqlite_account_info), 19
 start_large_file() (b2sdk.bucket.Bucket method), 32
 start_large_file() (b2sdk.raw_api.AbstractRawApi method), 58
 start_large_file() (b2sdk.raw_api.B2RawApi method), 59
 StorageCapExceeded, 43
 StreamWithHash (class in b2sdk.progress), 38
 subrange() (b2sdk.transferer.range.Range method), 68
 sync_folders() (in module b2sdk.sync.sync), 56
 SyncPolicyManager (class in b2sdk.sync.policy_manager), 51

T

take() (b2sdk.account_info.upload_url_pool.UploadUrlPool method), 21
 take_bucket_upload_url() (b2sdk.account_info.abstract.AbstractAccountInfo method), 18
 take_bucket_upload_url() (b2sdk.account_info.upload_url_pool.UrlPoolAccountInfo method), 22
 take_large_file_upload_url() (b2sdk.account_info.abstract.AbstractAccountInfo method), 18
 take_large_file_upload_url() (b2sdk.account_info.upload_url_pool.UrlPoolAccountInfo method), 22
 tell() (b2sdk.progress.AbstractStreamWithProgress method), 36
 TempDir (class in b2sdk.utils), 63
 test_http() (in module b2sdk.b2http), 63
 test_raw_api() (in module b2sdk.raw_api), 60
 test_raw_api_helper() (in module b2sdk.raw_api), 60
 TIMEOUT (b2sdk.b2http.B2Http attribute), 60
 TooManyRequests, 43
 TqdmProgressListener (class in b2sdk.progress), 39
 Transferer (class in b2sdk.transferer.transferer), 68
 TransientErrorMixin (class in b2sdk.exception), 43
 TruncatedOutput, 43

U

Unauthorized, 43
 UnexpectedCloudBehaviour, 43
 UnknownError, 43
 UnknownHost, 43
 unprintable_to_hex() (b2sdk.raw_api.B2RawApi method), 59
 UnrecognizedBucketType, 43
 UnsatisfiableRange, 44
 UnusableFileName, 44
 UpAndDeletePolicy (class in b2sdk.sync.policy), 53

UpAndKeepDaysPolicy (class in *b2sdk.sync.policy*),
 53
 update() (*b2sdk.bucket.Bucket* method), 32
 update_bucket() (*b2sdk.raw_api.AbstractRawApi*
 method), 58
 update_bucket() (*b2sdk.raw_api.B2RawApi*
 method), 59
 update_part_bytes()
 (*b2sdk.bucket.LargeFileUploadState* method),
 34
 upload() (*b2sdk.bucket.Bucket* method), 32
 upload_bytes() (*b2sdk.bucket.Bucket* method), 32
 upload_file() (*b2sdk.raw_api.B2RawApi* method),
 59
 upload_local_file() (*b2sdk.bucket.Bucket*
 method), 33
 upload_part() (*b2sdk.raw_api.AbstractRawApi*
 method), 58
 upload_part() (*b2sdk.raw_api.B2RawApi* method),
 60
 UploadSourceBytes (class in *b2sdk.upload_source*),
 24
 UploadSourceLocalFile (class in
 b2sdk.upload_source), 25
 UploadUrlPool (class in
 b2sdk.account_info.upload_url_pool), 21
 UpPolicy (class in *b2sdk.sync.policy*), 53
 url_for_api() (in module *b2sdk.api*), 28
 UrlPoolAccountInfo (class in
 b2sdk.account_info.upload_url_pool), 21

V

validate_b2_file_name() (in module
 b2sdk.utils), 65

W

write() (*b2sdk.progress.AbstractStreamWithProgress*
 method), 36
 write() (*b2sdk.progress.WritingStreamWithProgress*
 method), 39
 write_file_and_report_progress_context()
 (*b2sdk.download_dest.DownloadDestProgressWrapper*
 method), 24
 write_to_local_file_context()
 (*b2sdk.download_dest.DownloadDestLocalFile*
 method), 23
 write_to_local_file_context()
 (*b2sdk.download_dest.PreSeekedDownloadDest*
 method), 24
 WriterThread (class in *b2sdk.transferer.parallel*), 67
 WritingStreamWithProgress (class in
 b2sdk.progress), 39

Z

zip_folders() (in module *b2sdk.sync.sync*), 56