
Azure SDK for Python Documentation

Release 0.36.0

Microsoft

Aug 23, 2017

Contents

| | | |
|----------|--|------------|
| 1 | Installation: | 1 |
| 2 | Documentation: | 3 |
| 3 | Features: | 5 |
| 4 | System Requirements: | 7 |
| 5 | Need Help?: | 9 |
| 6 | Contributing: | 11 |
| 6.1 | Contribute Code or Provide Feedback: | 11 |
| 7 | Learn More | 13 |
| 8 | Indices and tables | 15 |
| 8.1 | Upgrade Guide | 15 |
| 8.2 | General Changes | 15 |
| 8.3 | Blob | 16 |
| 8.4 | Queue | 16 |
| 8.5 | Table | 16 |
| 8.6 | azure package | 16 |
| 8.7 | azure | 142 |
| | Python Module Index | 143 |

CHAPTER 1

Installation:

You can use `pip` to install the latest released version of `azure-storage`:

```
pip install azure-storage
```

If you want to install `azure-storage` from source:

```
git clone git://github.com/Azure/azure-storage-python.git
cd azure-storage-python
python setup.py install
```

If you are looking for Azure Service Bus or the Azure management libraries, please visit <https://github.com/Azure/azure-sdk-for-python>

CHAPTER 2

Documentation:

- [Blob Getting Started Doc – \(API\)](#)
- [Queue Getting Started Doc – \(API\)](#)
- [Table Getting Started Doc – \(API\)](#)
- [File Getting Started Doc – \(API\)](#)
- [Reference Documentation - All Services](#)

CHAPTER 3

Features:

- Blob
 - Create/Read/Update/Delete Containers
 - Create/Read/Update/Delete Blobs
 - Advanced Blob Operations
- Queue
 - Create/Delete Queues
 - Insert/Peek Queue Messages
 - Advanced Queue Operations
- Table
 - Create/Read/Update/Delete Tables
 - Create/Read/Update/Delete Entities
 - Batch operations
 - Advanced Table Operations
- Files
 - Create/Update/Delete Shares
 - Create/Update/Delete Directories
 - Create/Read/Update/Delete Files
 - Advanced File Operations

CHAPTER 4

System Requirements:

The supported Python versions are 2.7.x, 3.3.x, 3.4.x, and 3.5.x. To download Python, please visit <https://www.python.org/download/>

We recommend Python Tools for Visual Studio as a development environment for developing your applications. Please visit <http://aka.ms/python> for more information.

CHAPTER 5

Need Help?:

Be sure to check out the [Microsoft Azure Developer Forums on Stack Overflow](#) if you have trouble with the provided code.

Contributing:

Contribute Code or Provide Feedback:

If you would like to become an active contributor to this project, please follow the instructions provided in [Microsoft Azure Projects Contribution Guidelines](#).

If you encounter any bugs with the library, please file an issue in the [Issues](#) section of the project.

CHAPTER 7

[Learn More](#)

Microsoft Azure Python Developer Center

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

Upgrade Guide

This guide is intended to help upgrade code written for the Azure Storage Python library before version 0.30.0.

The best way to see how to upgrade a specific API is to take a look at the usage samples in the [Samples](#) directory on GitHub. A very specific set of changes as well as additions can be found in the [ChangeLog](#) and [BreakingChanges](#) documents. The below is a summary of those documents containing the most relevant pieces for the upgrade scenario.

General Changes

In general, we attempted to use more appropriate Python types for parameter and return values rather than always taking and receiving strings. Parameter and return values previously prefixed with `x_ms` were simplified by removing this prefix, and parameter values are divided by `'_'` between words as is idiomatic.

`Listing` returns a generator rather than a result segment. This generator automatically follows continuation tokens as more results are requested.

SAS methods take several individual parameters rather than a single parameter object. Similarly, ACL getters and setters take dictionaries mapping id to `AccessPolicy` rather than a list of `SignedIdentifiers` each holding an id and an `AccessPolicy`.

Blob

The single BlobService object was divided into three subservices for the different blob types (BlockBlobService, PageBlobService, AppendBlobService) with common methods in the abstract BaseBlobService these inherit from. This was done for two reasons. First, to reduce confusion when blob-type specific methods were used on the incorrect blob type. Second, to simplify the BlobService object which had grown to be quite extensive when append blob was added.

ContentSettings objects have replaced all content_* and cache_control parameters and return values for applicable APIs. This is intended to highlight that the semantics of setting content properties is replace rather than merge, meaning that setting any one property will cause any unspecified properties to be cleared on the blob.

On the performance side, single-threaded blob download APIs will now download the blob without chunking to improve perf and not do an initial get to find the size of the blob. However, as a result the progress_callback may receive None for its total parameter when parallelism is off to allow for this optimization.

Queue

The largest change to the QueueService class is that queue messages are both XML encoded and decoded by default. In past versions, either messages were not encoded or decoded by default, or only encoded. Encoding and decoding methods can be modified using the QueueService encode_function and decode_function instance variables.

Methods operating on queue messages and which have return values will return QueueMessage objects. QueueMessages objects contain essentially the same fields as previously, but the times are returned as dates rather than strings and the dequeue count is returned as an int.

Table

Rather than having a boolean switch for turning batching on and off, batches are an object which can be populated and then committed. Entities can be sent as dictionaries or as Entity objects, and returned entities are accessible as either objects or dictionaries. Methods which access and modify entites have been simplified so that if they take an entity object they extract the partition key and row key from that object rather than requiring these be sent separately.

All table entity integer values are stored on the service with type Edm.Int64 unless the type is explicitly overridden as Edm.Int32. Formerly, the type was decided based on the size of the number, but this resulted in hard to predict types on the service. So, the more consistent option was chosen.

Operations no longer echo content from the service and JSON is used instead of AtomPub, improving performance.

azure package

Subpackages

azure.storage package

Subpackages

azure.storage.blob package

Submodules

azure.storage.blob.appendblobservice module

```
class azure.storage.blob.appendblobservice.AppendBlobService (account_name=None,
                                                               account_key=None,
                                                               sas_token=None,
                                                               is_emulated=False,
                                                               protocol='https', end-
                                                               point_suffix='core.windows.net',
                                                               cus-
                                                               tom_domain=None,
                                                               re-
                                                               quest_session=None,
                                                               connec-
                                                               tion_string=None,
                                                               socket_timeout=None)
```

Bases: `azure.storage.blob.baseblobservice.BaseBlobService`

An append blob is comprised of blocks and is optimized for append operations. When you modify an append blob, blocks are added to the end of the blob only, via the `append_block` operation. Updating or deleting of existing blocks is not supported. Unlike a block blob, an append blob does not expose its block IDs.

Each block in an append blob can be a different size, up to a maximum of 4 MB, and an append blob can include up to 50,000 blocks. The maximum size of an append blob is therefore slightly more than 195 GB (4 MB X 50,000 blocks).

Variables `MAX_BLOCK_SIZE` (*int*) – The size of the blocks put by `append_blob_from_*` methods. Smaller blocks may be put if there is less data provided. The maximum block size the service supports is 4MB.

Parameters

- **account_name** (*str*) – The storage account name. This is used to authenticate requests signed with an account key and to construct the storage endpoint. It is required unless a connection string is given, or if a custom domain is used with anonymous authentication.
- **account_key** (*str*) – The storage account key. This is used for shared key authentication. If neither account key or sas token is specified, anonymous access will be used.
- **sas_token** (*str*) – A shared access signature token to use to authenticate requests instead of the account key. If account key and sas token are both specified, account key will be used to sign. If neither are specified, anonymous access will be used.
- **is_emulated** (*bool*) – Whether to use the emulator. Defaults to False. If specified, will override all other parameters besides connection string and request session.
- **protocol** (*str*) – The protocol to use for requests. Defaults to https.
- **endpoint_suffix** (*str*) – The host base component of the url, minus the account name. Defaults to Azure (core.windows.net). Override this to use the China cloud (core.chinacloudapi.cn).
- **custom_domain** (*str*) – The custom domain to use. This can be set in the Azure Portal. For example, 'www.mydomain.com'.
- **request_session** (*requests.Session*) – The session object to use for http requests.

- **connection_string** (*str*) – If specified, this will override all other parameters besides request session. See <http://azure.microsoft.com/en-us/documentation/articles/storage-configure-connection-string/> for the connection string format.
- **socket_timeout** (*int*) – If specified, this will override the default socket timeout. The timeout specified is in seconds. See DEFAULT_SOCKET_TIMEOUT in `_constants.py` for the default value.

MAX_BLOCK_SIZE = 4194304

append_blob_from_bytes (*container_name, blob_name, blob, index=0, count=None, validate_content=False, maxsize_condition=None, progress_callback=None, lease_id=None, timeout=None*)

Appends to the content of an existing blob from an array of bytes, with automatic chunking and progress notifications.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of blob to create or update.
- **blob** (*bytes*) – Content of blob as an array of bytes.
- **index** (*int*) – Start index in the array of bytes.
- **count** (*int*) – Number of bytes to upload. Set to None or negative value to upload all bytes starting from index.
- **validate_content** (*bool*) – If true, calculates an MD5 hash for each chunk of the blob. The storage service checks the hash of the content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the blob.
- **maxsize_condition** (*int*) – Optional conditional header. The max length in bytes permitted for the append blob. If the Append Block operation would cause the blob to exceed that limit or if the blob size is already greater than the value specified in this header, the request will fail with MaxBlobSizeConditionNotMet error (HTTP status code 412 - Precondition Failed).
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature function(current, total) where current is the number of bytes transferred so far, and total is the size of the blob, or None if the total size is unknown.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

Returns ETag and last modified properties for the Append Blob

Return type *ResourceProperties*

append_blob_from_path (*container_name, blob_name, file_path, validate_content=False, maxsize_condition=None, progress_callback=None, lease_id=None, timeout=None*)

Appends to the content of an existing blob from a file path, with automatic chunking and progress notifications.

Parameters

- **container_name** (*str*) – Name of existing container.

- **blob_name** (*str*) – Name of blob to create or update.
- **file_path** (*str*) – Path of the file to upload as the blob content.
- **validate_content** (*bool*) – If true, calculates an MD5 hash for each chunk of the blob. The storage service checks the hash of the content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the blob.
- **maxsize_condition** (*int*) – Optional conditional header. The max length in bytes permitted for the append blob. If the Append Block operation would cause the blob to exceed that limit or if the blob size is already greater than the value specified in this header, the request will fail with MaxBlobSizeConditionNotMet error (HTTP status code 412 - Precondition Failed).
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature function(current, total) where current is the number of bytes transferred so far, and total is the size of the blob, or None if the total size is unknown.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

Returns ETag and last modified properties for the Append Blob

Return type *ResourceProperties*

append_blob_from_stream(*container_name*, *blob_name*, *stream*, *count=None*,
validate_content=False, *maxsize_condition=None*,
progress_callback=None, *lease_id=None*, *timeout=None*)

Appends to the content of an existing blob from a file/stream, with automatic chunking and progress notifications.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of blob to create or update.
- **stream** (*io.IOBase*) – Opened stream to upload as the blob content.
- **count** (*int*) – Number of bytes to read from the stream. This is optional, but should be supplied for optimal performance.
- **validate_content** (*bool*) – If true, calculates an MD5 hash for each chunk of the blob. The storage service checks the hash of the content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the blob.
- **maxsize_condition** (*int*) – Conditional header. The max length in bytes permitted for the append blob. If the Append Block operation would cause the blob to exceed that limit or if the blob size is already greater than the value specified in this header, the request will fail with MaxBlobSizeConditionNotMet error (HTTP status code 412 - Precondition Failed).
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature function(current, total) where current is the number of bytes transferred so far, and total is the size of the blob, or None if the total size is unknown.

- **lease_id** (*str*) – Required if the blob has an active lease.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

Returns ETag and last modified properties for the Append Blob

Return type *ResourceProperties*

append_blob_from_text (*container_name*, *blob_name*, *text*, *encoding='utf-8'*, *validate_content=False*, *maxsize_condition=None*, *progress_callback=None*, *lease_id=None*, *timeout=None*)

Appends to the content of an existing blob from str/unicode, with automatic chunking and progress notifications.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of blob to create or update.
- **text** (*str*) – Text to upload to the blob.
- **encoding** (*str*) – Python encoding to use to convert the text to bytes.
- **validate_content** (*bool*) – If true, calculates an MD5 hash for each chunk of the blob. The storage service checks the hash of the content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the blob.
- **maxsize_condition** (*int*) – Optional conditional header. The max length in bytes permitted for the append blob. If the Append Block operation would cause the blob to exceed that limit or if the blob size is already greater than the value specified in this header, the request will fail with MaxBlobSizeConditionNotMet error (HTTP status code 412 - Precondition Failed).
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature function(current, total) where current is the number of bytes transferred so far, and total is the size of the blob, or None if the total size is unknown.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

Returns ETag and last modified properties for the Append Blob

Return type *ResourceProperties*

append_block (*container_name*, *blob_name*, *block*, *validate_content=False*, *maxsize_condition=None*, *appendpos_condition=None*, *lease_id=None*, *if_modified_since=None*, *if_unmodified_since=None*, *if_match=None*, *if_none_match=None*, *timeout=None*)

Commits a new block of data to the end of an existing append blob.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **block** (*bytes*) – Content of the block in bytes.

- **validate_content** (*bool*) – If true, calculates an MD5 hash of the block content. The storage service checks the hash of the content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the blob.
- **maxsize_condition** (*int*) – Optional conditional header. The max length in bytes permitted for the append blob. If the Append Block operation would cause the blob to exceed that limit or if the blob size is already greater than the value specified in this header, the request will fail with MaxBlobSizeConditionNotMet error (HTTP status code 412 - Precondition Failed).
- **appendpos_condition** (*int*) – Optional conditional header, used only for the Append Block operation. A number indicating the byte offset to compare. Append Block will succeed only if the append position is equal to this number. If it is not, the request will fail with the AppendPositionConditionNotMet error (HTTP status code 412 - Precondition Failed).
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns ETag, last modified, append offset, and committed block count properties for the updated Append Blob

Return type *AppendBlockProperties*

```
create_blob (container_name, blob_name, content_settings=None, metadata=None,
             lease_id=None, if_modified_since=None, if_unmodified_since=None, if_match=None,
             if_none_match=None, timeout=None)
```

Creates a blob or overrides an existing blob. Use `if_match=*` to prevent overriding an existing blob.

See `create_blob_from_*` for high level functions that handle the creation and upload of large blobs with automatic chunking and progress notifications.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of blob to create or update.

- **content_settings** (*ContentSettings*) – ContentSettings object used to set blob properties.
- **metadata** (*a dict mapping str to str*) – Name-value pairs associated with the blob as metadata.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns ETag and last modified properties for the updated Append Blob

Return type *ResourceProperties*

azure.storage.blob.baseblobservice module

```
class azure.storage.blob.baseblobservice.BaseBlobService (account_name=None,
                                                         account_key=None,
                                                         sas_token=None,
                                                         is_emulated=False,
                                                         protocol='https',      end-
                                                         point_suffix='core.windows.net',
                                                         custom_domain=None,
                                                         request_session=None,
                                                         connection_string=None,
                                                         socket_timeout=None)
```

Bases: *azure.storage.storageclient.StorageClient*

This is the main class managing Blob resources.

The Blob service stores text and binary data as blobs in the cloud. The Blob service offers the following three resources: the storage account, containers, and blobs. Within your storage account, containers provide a way to organize sets of blobs. For more information please see: <https://msdn.microsoft.com/en-us/library/azure/ee691964.aspx>

Variables

- **MAX_SINGLE_GET_SIZE** (*int*) – The size of the first range get performed by `get_blob_to_*` methods if `max_connections` is greater than 1. Less data will be returned if the blob is smaller than this.

- **MAX_CHUNK_GET_SIZE** (*int*) – The size of subsequent range gets performed by `get_blob_to_*` methods if `max_connections` is greater than 1 and the blob is larger than `MAX_SINGLE_GET_SIZE`. Less data will be returned if the remainder of the blob is smaller than this. If this is set to larger than 4MB, `content_validation` will throw an error if enabled. However, if `content_validation` is not desired a size greater than 4MB may be optimal. Setting this below 4MB is not recommended.
- **key_encryption_key** (*object*) – The key-encryption-key optionally provided by the user. If provided, will be used to encrypt/decrypt in supported methods. For methods requiring decryption, either the `key_encryption_key` OR the resolver must be provided. If both are provided, the resolver will take precedence. Must implement the following methods for APIs requiring encryption: `wrap_key(key)`—wraps the specified key (bytes) using an algorithm of the user’s choice. Returns the encrypted key as bytes. `get_key_wrap_algorithm()`—returns the algorithm used to wrap the specified symmetric key. `get_kid()`—returns a string key id for this key-encryption-key. Must implement the following methods for APIs requiring decryption: `unwrap_key(key, algorithm)`—returns the unwrapped form of the specified symmetric key using the string-specified algorithm. `get_kid()`—returns a string key id for this key-encryption-key.
- **key_resolver_function(kid)** (*function*) – A function to resolve keys optionally provided by the user. If provided, will be used to decrypt in supported methods. For methods requiring decryption, either the `key_encryption_key` OR the resolver must be provided. If both are provided, the resolver will take precedence. It uses the `kid` string to return a key-encryption-key implementing the interface defined above.
- **require_encryption** (*bool*) – A flag that may be set to ensure that all messages successfully uploaded to the queue and all those downloaded and successfully read from the queue are/were encrypted while on the server. If this flag is set, all required parameters for encryption/decryption must be provided. See the above comments on the `key_encryption_key` and `resolver`.

Parameters

- **account_name** (*str*) – The storage account name. This is used to authenticate requests signed with an account key and to construct the storage endpoint. It is required unless a connection string is given, or if a custom domain is used with anonymous authentication.
- **account_key** (*str*) – The storage account key. This is used for shared key authentication. If neither account key or sas token is specified, anonymous access will be used.
- **sas_token** (*str*) – A shared access signature token to use to authenticate requests instead of the account key. If account key and sas token are both specified, account key will be used to sign. If neither are specified, anonymous access will be used.
- **is_emulated** (*bool*) – Whether to use the emulator. Defaults to `False`. If specified, will override all other parameters besides connection string and request session.
- **protocol** (*str*) – The protocol to use for requests. Defaults to `https`.
- **endpoint_suffix** (*str*) – The host base component of the url, minus the account name. Defaults to `Azure (core.windows.net)`. Override this to use the China cloud (`core.chinacloudapi.cn`).
- **custom_domain** (*str*) – The custom domain to use. This can be set in the Azure Portal. For example, `'www.mydomain.com'`.
- **request_session** (*requests.Session*) – The session object to use for http requests.

- **connection_string** (*str*) – If specified, this will override all other parameters besides request session. See <http://azure.microsoft.com/en-us/documentation/articles/storage-configure-connection-string/> for the connection string format
- **socket_timeout** (*int*) – If specified, this will override the default socket timeout. The timeout specified is in seconds. See `DEFAULT_SOCKET_TIMEOUT` in `_constants.py` for the default value.

MAX_CHUNK_GET_SIZE = 4194304

MAX_SINGLE_GET_SIZE = 33554432

abort_copy_blob (*container_name, blob_name, copy_id, lease_id=None, timeout=None*)

Aborts a pending copy_blob operation, and leaves a destination blob with zero length and full metadata.

Parameters

- **container_name** (*str*) – Name of destination container.
- **blob_name** (*str*) – Name of destination blob.
- **copy_id** (*str*) – Copy identifier provided in the copy.id of the original copy_blob operation.
- **lease_id** (*str*) – Required if the destination blob has an active infinite lease.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

acquire_blob_lease (*container_name, blob_name, lease_duration=-1, proposed_lease_id=None, if_modified_since=None, if_unmodified_since=None, if_match=None, if_none_match=None, timeout=None*)

Requests a new lease. If the blob does not have an active lease, the Blob service creates a lease on the blob and returns a new lease ID.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **lease_duration** (*int*) – Specifies the duration of the lease, in seconds, or negative one (-1) for a lease that never expires. A non-infinite lease can be between 15 and 60 seconds. A lease duration cannot be changed using renew or change. Default is -1 (infinite lease).
- **proposed_lease_id** (*str*) – Proposed lease ID, in a GUID string format. The Blob service returns 400 (Invalid request) if the proposed lease ID is not in the correct format.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag matches the value specified.

- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns the lease ID of the newly created lease.

Returns *str*

acquire_container_lease (*container_name*, *lease_duration=-1*, *proposed_lease_id=None*, *if_modified_since=None*, *if_unmodified_since=None*, *timeout=None*)

Requests a new lease. If the container does not have an active lease, the Blob service creates a lease on the container and returns a new lease ID.

Parameters

- **container_name** (*str*) – Name of existing container.
- **lease_duration** (*int*) – Specifies the duration of the lease, in seconds, or negative one (-1) for a lease that never expires. A non-infinite lease can be between 15 and 60 seconds. A lease duration cannot be changed using renew or change. Default is -1 (infinite lease).
- **proposed_lease_id** (*str*) – Proposed lease ID, in a GUID string format. The Blob service returns 400 (Invalid request) if the proposed lease ID is not in the correct format.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns the lease ID of the newly created lease.

Returns *str*

break_blob_lease (*container_name*, *blob_name*, *lease_break_period=None*, *if_modified_since=None*, *if_unmodified_since=None*, *if_match=None*, *if_none_match=None*, *timeout=None*)

Breaks the lease, if the blob has an active lease. Once a lease is broken, it cannot be renewed. Any authorized request can break the lease; the request is not required to specify a matching lease ID. When a lease is broken, the lease break period is allowed to elapse, during which time no lease operation except break and release can be performed on the blob. When a lease is successfully broken, the response indicates the interval in seconds until a new lease can be acquired.

A lease that has been broken can also be released, in which case another client may immediately acquire the lease on the blob.

Parameters

- **container_name** (*str*) – Name of existing container.

- **blob_name** (*str*) – Name of existing blob.
- **lease_break_period** (*int*) – For a break operation, this is the proposed duration of seconds that the lease should continue before it is broken, between 0 and 60 seconds. This break period is only used if it is shorter than the time remaining on the lease. If longer, the time remaining on the lease is used. A new lease will not be available before the break period has expired, but the lease may be held for longer than the break period. If this header does not appear with a break operation, a fixed-duration lease breaks after the remaining lease period elapses, and an infinite lease breaks immediately.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns Approximate time remaining in the lease period, in seconds.

Returns int

break_container_lease (*container_name*, *lease_break_period=None*, *if_modified_since=None*, *if_unmodified_since=None*, *timeout=None*)

Break the lease, if the container has an active lease. Once a lease is broken, it cannot be renewed. Any authorized request can break the lease; the request is not required to specify a matching lease ID. When a lease is broken, the lease break period is allowed to elapse, during which time no lease operation except break and release can be performed on the container. When a lease is successfully broken, the response indicates the interval in seconds until a new lease can be acquired.

Parameters

- **container_name** (*str*) – Name of existing container.
- **lease_break_period** (*int*) – This is the proposed duration of seconds that the lease should continue before it is broken, between 0 and 60 seconds. This break period is only used if it is shorter than the time remaining on the lease. If longer, the time remaining on the lease is used. A new lease will not be available before the break period has expired, but the lease may be held for longer than the break period. If this header does not appear with a break operation, a fixed-duration lease breaks after the remaining lease period elapses, and an infinite lease breaks immediately.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.

- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns Approximate time remaining in the lease period, in seconds.

Returns int

```
change_blob_lease (container_name, blob_name, lease_id, proposed_lease_id,
                    if_modified_since=None, if_unmodified_since=None, if_match=None,
                    if_none_match=None, timeout=None)
```

Changes the lease ID of an active lease. A change must include the current lease ID and a new lease ID.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **proposed_lease_id** (*str*) – Proposed lease ID, in a GUID string format. The Blob service returns 400 (Invalid request) if the proposed lease ID is not in the correct format.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

```
change_container_lease (container_name, lease_id, proposed_lease_id,
                          if_modified_since=None, if_unmodified_since=None, timeout=None)
```

Change the lease ID of an active lease. A change must include the current lease ID and a new lease ID.

Parameters

- **container_name** (*str*) – Name of existing container.
- **lease_id** (*str*) – Lease ID for active lease.
- **proposed_lease_id** (*str*) – Proposed lease ID, in a GUID string format. The Blob service returns 400 (Invalid request) if the proposed lease ID is not in the correct format.

- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

copy_blob (*container_name*, *blob_name*, *copy_source*, *metadata=None*, *source_if_modified_since=None*, *source_if_unmodified_since=None*, *source_if_match=None*, *source_if_none_match=None*, *destination_if_modified_since=None*, *destination_if_unmodified_since=None*, *destination_if_match=None*, *destination_if_none_match=None*, *destination_lease_id=None*, *source_lease_id=None*, *timeout=None*)

Copies a blob asynchronously. This operation returns a copy operation properties object, including a copy ID you can use to check or abort the copy operation. The Blob service copies blobs on a best-effort basis.

The source blob for a copy operation may be a block blob, an append blob, or a page blob. If the destination blob already exists, it must be of the same blob type as the source blob. Any existing destination blob will be overwritten. The destination blob cannot be modified while a copy operation is in progress.

When copying from a page blob, the Blob service creates a destination page blob of the source blob's length, initially containing all zeroes. Then the source page ranges are enumerated, and non-empty ranges are copied.

For a block blob or an append blob, the Blob service creates a committed blob of zero length before returning from this operation. When copying from a block blob, all committed blocks and their block IDs are copied. Uncommitted blocks are not copied. At the end of the copy operation, the destination blob will have the same committed block count as the source.

When copying from an append blob, all committed blocks are copied. At the end of the copy operation, the destination blob will have the same committed block count as the source.

For all blob types, you can call `get_blob_properties` on the destination blob to check the status of the copy operation. The final blob will be committed when the copy completes.

Parameters

- **container_name** (*str*) – Name of the destination container. The container must exist.
- **blob_name** (*str*) – Name of the destination blob. If the destination blob exists, it will be overwritten. Otherwise, it will be created.
- **copy_source** (*str*) – A URL of up to 2 KB in length that specifies an Azure file or blob. The value should be URL-encoded as it would appear in a request URI. If the source is in another account, the source must either be public or must be authenticated via a shared access signature. If the source is public, no authentication is required. Examples: <https://myaccount.blob.core.windows.net/mycontainer/myblob> <https://myaccount.blob.core.windows.net/mycontainer/myblob?snapshot=<DateTime>> <https://otheraccount.blob.core.windows.net/mycontainer/myblob?sastoken>
- **metadata** (*A dict mapping str to str.*) – Name-value pairs associated with the blob as metadata. If no name-value pairs are specified, the operation will copy the metadata from the source blob or file to the destination blob. If one or more name-value

pairs are specified, the destination blob is created with the specified metadata, and metadata is not copied from the source blob or file.

- **source_if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this conditional header to copy the blob only if the source blob has been modified since the specified date/time.
- **source_if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this conditional header to copy the blob only if the source blob has not been modified since the specified date/time.
- **source_if_match** (*ETag*) – An ETag value, or the wildcard character (*). Specify this conditional header to copy the source blob only if its ETag matches the value specified. If the ETag values do not match, the Blob service returns status code 412 (Precondition Failed). This header cannot be specified if the source is an Azure File.
- **source_if_none_match** (*ETag*) – An ETag value, or the wildcard character (*). Specify this conditional header to copy the blob only if its ETag does not match the value specified. If the values are identical, the Blob service returns status code 412 (Precondition Failed). This header cannot be specified if the source is an Azure File.
- **destination_if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this conditional header to copy the blob only if the destination blob has been modified since the specified date/time. If the destination blob has not been modified, the Blob service returns status code 412 (Precondition Failed).
- **destination_if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this conditional header to copy the blob only if the destination blob has not been modified since the specified date/time. If the destination blob has been modified, the Blob service returns status code 412 (Precondition Failed).
- **destination_if_match** (*ETag*) – An ETag value, or the wildcard character (*). Specify an ETag value for this conditional header to copy the blob only if the specified ETag value matches the ETag value for an existing destination blob. If the ETag for the destination blob does not match the ETag specified for If-Match, the Blob service returns status code 412 (Precondition Failed).
- **destination_if_none_match** (*ETag*) – An ETag value, or the wildcard character (*). Specify an ETag value for this conditional header to copy the blob only if the specified ETag value does not match the ETag value for the destination blob. Specify the wildcard character (*) to perform the operation only if the destination blob does not exist. If the specified condition isn't met, the Blob service returns status code 412 (Precondition Failed).
- **destination_lease_id** (*str*) – The lease ID specified for this header must match the lease ID of the destination blob. If the request does not include the lease ID or it is not valid, the operation fails with status code 412 (Precondition Failed).
- **source_lease_id** (*str*) – Specify this to perform the Copy Blob operation only if the lease ID given matches the active lease ID of the source blob.

- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns Copy operation properties such as status, source, and ID.

Return type *CopyProperties*

create_container (*container_name*, *metadata=None*, *public_access=None*, *fail_on_exist=False*, *timeout=None*)

Creates a new container under the specified account. If the container with the same name already exists, the operation fails if *fail_on_exist* is True.

Parameters

- **container_name** (*str*) – Name of container to create.
- **metadata** (*a dict mapping str to str*) – A dict with name_value pairs to associate with the container as metadata. Example: { 'Category': 'test' }
- **public_access** (One of the values listed in the *PublicAccess* enum.) – Possible values include: container, blob.
- **fail_on_exist** (*bool*) – Specify whether to throw an exception when the container exists.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns True if container is created, False if container already exists.

Return type bool

delete_blob (*container_name*, *blob_name*, *snapshot=None*, *lease_id=None*, *delete_snapshots=None*, *if_modified_since=None*, *if_unmodified_since=None*, *if_match=None*, *if_none_match=None*, *timeout=None*)

Marks the specified blob or snapshot for deletion. The blob is later deleted during garbage collection.

Note that in order to delete a blob, you must delete all of its snapshots. You can delete both at the same time with the Delete Blob operation.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **snapshot** (*str*) – The snapshot parameter is an opaque DateTime value that, when present, specifies the blob snapshot to delete.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **delete_snapshots** (One of the values listed in the *DeleteSnapshot* enum.) – Required if the blob has associated snapshots.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.

- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

delete_container (*container_name*, *fail_not_exist=False*, *lease_id=None*, *if_modified_since=None*, *if_unmodified_since=None*, *timeout=None*)

Marks the specified container for deletion. The container and any blobs contained within it are later deleted during garbage collection.

Parameters

- **container_name** (*str*) – Name of container to delete.
- **fail_not_exist** (*bool*) – Specify whether to throw an exception when the container doesn’t exist.
- **lease_id** (*str*) – If specified, delete_container only succeeds if the container’s lease is active and matches this ID. Required if the container has an active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns True if container is deleted, False container doesn’t exist.

Return type bool

exists (*container_name*, *blob_name=None*, *snapshot=None*, *timeout=None*)

Returns a boolean indicating whether the container exists (if blob_name is None), or otherwise a boolean indicating whether the blob exists.

Parameters

- **container_name** (*str*) – Name of a container.
- **blob_name** (*str*) – Name of a blob. If None, the container will be checked for existence.
- **snapshot** (*str*) – The snapshot parameter is an opaque DateTime value that, when present, specifies the snapshot.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns A boolean indicating whether the resource exists.

Return type bool

generate_account_shared_access_signature (*resource_types, permission, expiry, start=None, ip=None, protocol=None*)

Generates a shared access signature for the blob service. Use the returned signature with the `sas_token` parameter of any `BlobService`.

Parameters

- **resource_types** (`ResourceTypes`) – Specifies the resource types that are accessible with the account SAS.
- **permission** (`AccountPermissions`) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy.
- **expiry** (`datetime.datetime` or `str`) – The time at which the shared access signature becomes invalid. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **start** (`datetime.datetime` or `str`) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **ip** (`str`) – Specifies an IP address or a range of IP addresses from which to accept requests. If the IP address from which the request originates does not match the IP address or address range specified on the SAS token, the request is not authenticated. For example, specifying `sip=168.1.5.65` or `sip=168.1.5.60-168.1.5.70` on the SAS restricts the request to those IP addresses.
- **protocol** (`str`) – Specifies the protocol permitted for a request made. The default value is `https,http`. See *Protocol* for possible values.

Returns A Shared Access Signature (sas) token.

Return type `str`

generate_blob_shared_access_signature (*container_name, blob_name, permission=None, expiry=None, start=None, id=None, ip=None, protocol=None, cache_control=None, content_disposition=None, content_encoding=None, content_language=None, content_type=None*)

Generates a shared access signature for the blob. Use the returned signature with the `sas_token` parameter of any `BlobService`.

Parameters

- **container_name** (`str`) – Name of container.
- **blob_name** (`str`) – Name of blob.
- **permission** (`BlobPermissions`) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Permissions must be ordered read, write, delete, list. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy.
- **expiry** (`datetime.datetime` or `str`) – The time at which the shared access signature becomes invalid. Required unless an id is given referencing a stored access policy

which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.

- **start** (*datetime.datetime* or *str*) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **id** (*str*) – A unique value up to 64 characters in length that correlates to a stored access policy. To create a stored access policy, use `set_container_acl()`.
- **ip** (*str*) – Specifies an IP address or a range of IP addresses from which to accept requests. If the IP address from which the request originates does not match the IP address or address range specified on the SAS token, the request is not authenticated. For example, specifying `sip=168.1.5.65` or `sip=168.1.5.60-168.1.5.70` on the SAS restricts the request to those IP addresses.
- **protocol** (*str*) – Specifies the protocol permitted for a request made. The default value is `https,http`. See *Protocol* for possible values.
- **cache_control** (*str*) – Response header value for Cache-Control when resource is accessed using this shared access signature.
- **content_disposition** (*str*) – Response header value for Content-Disposition when resource is accessed using this shared access signature.
- **content_encoding** (*str*) – Response header value for Content-Encoding when resource is accessed using this shared access signature.
- **content_language** (*str*) – Response header value for Content-Language when resource is accessed using this shared access signature.
- **content_type** (*str*) – Response header value for Content-Type when resource is accessed using this shared access signature.

Returns A Shared Access Signature (sas) token.

Return type `str`

```
generate_container_shared_access_signature(container_name,      permission=None,
                                           expiry=None,        start=None,
                                           id=None,           ip=None,      proto-
                                           col=None,          cache_control=None,
                                           content_disposition=None,  con-
                                           tent_encoding=None,        con-
                                           tent_language=None,      con-
                                           tent_type=None)
```

Generates a shared access signature for the container. Use the returned signature with the `sas_token` parameter of any `BlobService`.

Parameters

- **container_name** (*str*) – Name of container.
- **permission** (`ContainerPermissions`) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Permissions must be ordered read, write, delete, list. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy.

- **expiry** (*datetime.datetime or str*) – The time at which the shared access signature becomes invalid. Required unless an `id` is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **start** (*datetime.datetime or str*) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **id** (*str*) – A unique value up to 64 characters in length that correlates to a stored access policy. To create a stored access policy, use `set_blob_service_properties`.
- **ip** (*str*) – Specifies an IP address or a range of IP addresses from which to accept requests. If the IP address from which the request originates does not match the IP address or address range specified on the SAS token, the request is not authenticated. For example, specifying `sip=168.1.5.65` or `sip=168.1.5.60-168.1.5.70` on the SAS restricts the request to those IP addresses.
- **protocol** (*str*) – Specifies the protocol permitted for a request made. The default value is `https,http`. See *Protocol* for possible values.
- **cache_control** (*str*) – Response header value for Cache-Control when resource is accessed using this shared access signature.
- **content_disposition** (*str*) – Response header value for Content-Disposition when resource is accessed using this shared access signature.
- **content_encoding** (*str*) – Response header value for Content-Encoding when resource is accessed using this shared access signature.
- **content_language** (*str*) – Response header value for Content-Language when resource is accessed using this shared access signature.
- **content_type** (*str*) – Response header value for Content-Type when resource is accessed using this shared access signature.

Returns A Shared Access Signature (sas) token.

Return type `str`

```
get_blob_metadata (container_name, blob_name, snapshot=None, lease_id=None,
                    if_modified_since=None, if_unmodified_since=None, if_match=None,
                    if_none_match=None, timeout=None)
```

Returns all user-defined metadata for the specified blob or snapshot.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **snapshot** (*str*) – The snapshot parameter is an opaque value that, when present, specifies the blob snapshot to retrieve.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A `DateTime` value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.

- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns A dictionary representing the blob metadata name, value pairs.

Return type a dict mapping str to str

get_blob_properties (*container_name*, *blob_name*, *snapshot=None*, *lease_id=None*, *if_modified_since=None*, *if_unmodified_since=None*, *if_match=None*, *if_none_match=None*, *timeout=None*)

Returns all user-defined metadata, standard HTTP properties, and system properties for the blob. It does not return the content of the blob. Returns *Blob* with *BlobProperties* and a metadata dict.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **snapshot** (*str*) – The snapshot parameter is an opaque DateTime value that, when present, specifies the blob snapshot to retrieve.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns a blob object including properties and metadata.

Return type *Blob*

get_blob_service_properties (*timeout=None*)

Gets the properties of a storage account's Blob service, including Azure Storage Analytics.

Parameters **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns The blob service properties.

Return type *ServiceProperties* with an attached `target_version` property

get_blob_service_stats (*timeout=None*)

Retrieves statistics related to replication for the Blob service. It is only available when read-access geo-redundant replication is enabled for the storage account.

With geo-redundant replication, Azure Storage maintains your data durable in two locations. In both locations, Azure Storage constantly maintains multiple healthy replicas of your data. The location where you read, create, update, or delete data is the primary storage account location. The primary location exists in the region you choose at the time you create an account via the Azure Management Azure classic portal, for example, North Central US. The location to which your data is replicated is the secondary location. The secondary location is automatically determined based on the location of the primary; it is in a second data center that resides in the same region as the primary location. Read-only access is available from the secondary location, if read-access geo-redundant replication is enabled for your storage account.

Parameters **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns The blob service stats.

Return type *ServiceStats*

get_blob_to_bytes (*container_name, blob_name, snapshot=None, start_range=None, end_range=None, validate_content=False, progress_callback=None, max_connections=2, lease_id=None, if_modified_since=None, if_unmodified_since=None, if_match=None, if_none_match=None, timeout=None*)

Downloads a blob as an array of bytes, with automatic chunking and progress notifications. Returns an instance of `Blob` with properties, metadata, and content.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **snapshot** (*str*) – The snapshot parameter is an opaque `DateTime` value that, when present, specifies the blob snapshot to retrieve.
- **start_range** (*int*) – Start of byte range to use for downloading a section of the blob. If no `end_range` is given, all bytes after the `start_range` will be downloaded. The `start_range` and `end_range` params are inclusive. Ex: `start_range=0, end_range=511` will download first 512 bytes of blob.
- **end_range** (*int*) – End of byte range to use for downloading a section of the blob. If `end_range` is given, `start_range` must be provided. The `start_range` and `end_range` params are inclusive. Ex: `start_range=0, end_range=511` will download first 512 bytes of blob.
- **validate_content** (*bool*) – If set to true, validates an MD5 hash for each retrieved portion of the blob. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that the service will only return transactional MD5s for chunks 4MB or less so the first get request will be of size `self.MAX_CHUNK_GET_SIZE` instead of `self.MAX_SINGLE_GET_SIZE`. If `self.MAX_CHUNK_GET_SIZE` was set to greater than 4MB an error will be thrown. As computing the MD5 takes processing time and more requests will need to be done due to the reduced chunk size there may be some increase in latency.

- **progress_callback** (*callback function in format of `func(current, total)`*) – Callback for progress with signature `function(current, total)` where `current` is the number of bytes transferred so far, and `total` is the size of the blob if known.
- **max_connections** (*int*) – If set to 2 or greater, an initial get will be done for the first `self.MAX_SINGLE_GET_SIZE` bytes of the blob. If this is the entire blob, the method returns at this point. If it is not, it will download the remaining data parallel using the number of threads equal to `max_connections`. Each chunk will be of size `self.MAX_CHUNK_GET_SIZE`. If set to 1, a single large get request will be done. This is not generally recommended but available if very few threads should be used, network requests are very expensive, or a non-seekable stream prevents parallel download. This may also be useful if many blobs are expected to be empty as an extra request is required for empty blobs if `max_connections` is greater than 1.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A `DateTime` value. Azure expects the date value passed in to be UTC. If `timezone` is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without `timezone` info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A `DateTime` value. Azure expects the date value passed in to be UTC. If `timezone` is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without `timezone` info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

Returns A `Blob` with properties and metadata. If `max_connections` is greater than 1, the `content_md5` (if set on the blob) will not be returned. If you require this value, either use `get_blob_properties` or set `max_connections` to 1.

Return type `Blob`

```
get_blob_to_path(container_name, blob_name, file_path, open_mode='wb', snapshot=None, start_range=None, end_range=None, validate_content=False, progress_callback=None, max_connections=2, lease_id=None, if_modified_since=None, if_unmodified_since=None, if_match=None, if_none_match=None, timeout=None)
```

Downloads a blob to a file path, with automatic chunking and progress notifications. Returns an instance of `Blob` with properties and metadata.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **file_path** (*str*) – Path of file to write out to.

- **open_mode** (*str*) – Mode to use when opening the file. Note that specifying append only open_mode prevents parallel download. So, max_connections must be set to 1 if this open_mode is used.
- **snapshot** (*str*) – The snapshot parameter is an opaque DateTime value that, when present, specifies the blob snapshot to retrieve.
- **start_range** (*int*) – Start of byte range to use for downloading a section of the blob. If no end_range is given, all bytes after the start_range will be downloaded. The start_range and end_range params are inclusive. Ex: start_range=0, end_range=511 will download first 512 bytes of blob.
- **end_range** (*int*) – End of byte range to use for downloading a section of the blob. If end_range is given, start_range must be provided. The start_range and end_range params are inclusive. Ex: start_range=0, end_range=511 will download first 512 bytes of blob.
- **validate_content** (*bool*) – If set to true, validates an MD5 hash for each retrieved portion of the blob. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that the service will only return transactional MD5s for chunks 4MB or less so the first get request will be of size self.MAX_CHUNK_GET_SIZE instead of self.MAX_SINGLE_GET_SIZE. If self.MAX_CHUNK_GET_SIZE was set to greater than 4MB an error will be thrown. As computing the MD5 takes processing time and more requests will need to be done due to the reduced chunk size there may be some increase in latency.
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature function(current, total) where current is the number of bytes transfered so far, and total is the size of the blob if known.
- **max_connections** (*int*) – If set to 2 or greater, an initial get will be done for the first self.MAX_SINGLE_GET_SIZE bytes of the blob. If this is the entire blob, the method returns at this point. If it is not, it will download the remaining data parallel using the number of threads equal to max_connections. Each chunk will be of size self.MAX_CHUNK_GET_SIZE. If set to 1, a single large get request will be done. This is not generally recommended but available if very few threads should be used, network requests are very expensive, or a non-seekable stream prevents parallel download. This may also be useful if many blobs are expected to be empty as an extra request is required for empty blobs if max_connections is greater than 1.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag does not match the value

specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.

- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

Returns A Blob with properties and metadata. If `max_connections` is greater than 1, the `content_md5` (if set on the blob) will not be returned. If you require this value, either use `get_blob_properties` or set `max_connections` to 1.

Return type *Blob*

get_blob_to_stream(*container_name, blob_name, stream, snapshot=None, start_range=None, end_range=None, validate_content=False, progress_callback=None, max_connections=2, lease_id=None, if_modified_since=None, if_unmodified_since=None, if_match=None, if_none_match=None, timeout=None*)

Downloads a blob to a stream, with automatic chunking and progress notifications. Returns an instance of Blob with properties and metadata.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **stream** (*io.IOBase*) – Opened stream to write to.
- **snapshot** (*str*) – The snapshot parameter is an opaque DateTime value that, when present, specifies the blob snapshot to retrieve.
- **start_range** (*int*) – Start of byte range to use for downloading a section of the blob. If no `end_range` is given, all bytes after the `start_range` will be downloaded. The `start_range` and `end_range` params are inclusive. Ex: `start_range=0, end_range=511` will download first 512 bytes of blob.
- **end_range** (*int*) – End of byte range to use for downloading a section of the blob. If `end_range` is given, `start_range` must be provided. The `start_range` and `end_range` params are inclusive. Ex: `start_range=0, end_range=511` will download first 512 bytes of blob.
- **validate_content** (*bool*) – If set to true, validates an MD5 hash for each retrieved portion of the blob. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that the service will only return transactional MD5s for chunks 4MB or less so the first get request will be of size `self.MAX_CHUNK_GET_SIZE` instead of `self.MAX_SINGLE_GET_SIZE`. If `self.MAX_CHUNK_GET_SIZE` was set to greater than 4MB an error will be thrown. As computing the MD5 takes processing time and more requests will need to be done due to the reduced chunk size there may be some increase in latency.
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature `function(current, total)` where `current` is the number of bytes transferred so far, and `total` is the size of the blob if known.
- **max_connections** (*int*) – If set to 2 or greater, an initial get will be done for the first `self.MAX_SINGLE_GET_SIZE` bytes of the blob. If this is the entire blob, the method returns at this point. If it is not, it will download the remaining data parallel using the number of threads equal to `max_connections`. Each chunk will be of size `self.MAX_CHUNK_GET_SIZE`. If set to 1, a single large get request will be done. This is not generally recommended but available if very few threads should be used, network requests are very expensive, or a non-seekable stream prevents parallel download. This

may also be useful if many blobs are expected to be empty as an extra request is required for empty blobs if `max_connections` is greater than 1.

- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

Returns A Blob with properties and metadata. If `max_connections` is greater than 1, the `content_md5` (if set on the blob) will not be returned. If you require this value, either use `get_blob_properties` or set `max_connections` to 1.

Return type *Blob*

```
get_blob_to_text (container_name, blob_name, encoding='utf-8', snapshot=None,
                 start_range=None, end_range=None, validate_content=False,
                 progress_callback=None, max_connections=2, lease_id=None,
                 if_modified_since=None, if_unmodified_since=None, if_match=None,
                 if_none_match=None, timeout=None)
```

Downloads a blob as unicode text, with automatic chunking and progress notifications. Returns an instance of `Blob` with properties, metadata, and content.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **encoding** (*str*) – Python encoding to use when decoding the blob data.
- **snapshot** (*str*) – The snapshot parameter is an opaque DateTime value that, when present, specifies the blob snapshot to retrieve.
- **start_range** (*int*) – Start of byte range to use for downloading a section of the blob. If no `end_range` is given, all bytes after the `start_range` will be downloaded. The `start_range` and `end_range` params are inclusive. Ex: `start_range=0, end_range=511` will download first 512 bytes of blob.
- **end_range** (*int*) – End of byte range to use for downloading a section of the blob. If `end_range` is given, `start_range` must be provided. The `start_range` and `end_range` params are inclusive. Ex: `start_range=0, end_range=511` will download first 512 bytes of blob.

- **validate_content** (*bool*) – If set to true, validates an MD5 hash for each retrieved portion of the blob. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that the service will only return transactional MD5s for chunks 4MB or less so the first get request will be of size `self.MAX_CHUNK_GET_SIZE` instead of `self.MAX_SINGLE_GET_SIZE`. If `self.MAX_CHUNK_GET_SIZE` was set to greater than 4MB an error will be thrown. As computing the MD5 takes processing time and more requests will need to be done due to the reduced chunk size there may be some increase in latency.
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature `function(current, total)` where `current` is the number of bytes transfered so far, and `total` is the size of the blob if known.
- **max_connections** (*int*) – If set to 2 or greater, an initial get will be done for the first `self.MAX_SINGLE_GET_SIZE` bytes of the blob. If this is the entire blob, the method returns at this point. If it is not, it will download the remaining data parallel using the number of threads equal to `max_connections`. Each chunk will be of size `self.MAX_CHUNK_GET_SIZE`. If set to 1, a single large get request will be done. This is not generally recommended but available if very few threads should be used, network requests are very expensive, or a non-seekable stream prevents parallel download. This may also be useful if many blobs are expected to be empty as an extra request is required for empty blobs if `max_connections` is greater than 1.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If `timezone` is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without `timezone` info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If `timezone` is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without `timezone` info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

Returns A Blob with properties and metadata. If `max_connections` is greater than 1, the `content_md5` (if set on the blob) will not be returned. If you require this value, either use `get_blob_properties` or set `max_connections` to 1.

Return type *Blob*

get_container_acl (*container_name, lease_id=None, timeout=None*)

Gets the permissions for the specified container. The permissions indicate whether container data may be accessed publicly.

Parameters

- **container_name** (*str*) – Name of existing container.
- **lease_id** – If specified, `get_container_acl` only succeeds if the container’s lease is active and matches this ID.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns A dictionary of access policies associated with the container.

Return type dict of str to *AccessPolicy* and a `public_access` property if public access is turned on

get_container_metadata (*container_name*, *lease_id=None*, *timeout=None*)

Returns all user-defined metadata for the specified container.

Parameters

- **container_name** (*str*) – Name of existing container.
- **lease_id** (*str*) – If specified, `get_container_metadata` only succeeds if the container’s lease is active and matches this ID.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns A dictionary representing the container metadata name, value pairs.

Return type a dict mapping str to str

get_container_properties (*container_name*, *lease_id=None*, *timeout=None*)

Returns all user-defined metadata and system properties for the specified container. The data returned does not include the container’s list of blobs.

Parameters

- **container_name** (*str*) – Name of existing container.
- **lease_id** (*str*) – If specified, `get_container_properties` only succeeds if the container’s lease is active and matches this ID.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns properties for the specified container within a container object.

Return type *Container*

list_blobs (*container_name*, *prefix=None*, *num_results=None*, *include=None*, *delimiter=None*, *marker=None*, *timeout=None*)

Returns a generator to list the blobs under the specified container. The generator will lazily follow the continuation tokens returned by the service and stop when all blobs have been returned or `num_results` is reached.

If `num_results` is specified and the account has more than that number of blobs, the generator will have a populated `next_marker` field once it finishes. This marker can be used to create a new generator if more results are desired.

Parameters

- **container_name** (*str*) – Name of existing container.
- **prefix** (*str*) – Filters the results to return only blobs whose names begin with the specified prefix.
- **num_results** (*int*) – Specifies the maximum number of blobs to return, including all `BlobPrefix` elements. If the request does not specify `num_results` or specifies a value greater than 5,000, the server will return up to 5,000 items. Setting `num_results` to a value less than or equal to zero results in error response code 400 (Bad Request).

- **include** (*Include*) – Specifies one or more additional datasets to include in the response.
- **delimiter** (*str*) – When the request includes this parameter, the operation returns a *BlobPrefix* element in the result list that acts as a placeholder for all blobs whose names begin with the same substring up to the appearance of the delimiter character. The delimiter may be a single character or a string.
- **marker** (*str*) – An opaque continuation token. This value can be retrieved from the *next_marker* field of a previous generator object if *num_results* was specified and that generator has finished enumerating results. If specified, this generator will begin returning results from the point where the previous generator stopped.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

list_containers (*prefix=None, num_results=None, include_metadata=False, marker=None, timeout=None*)

Returns a generator to list the containers under the specified account. The generator will lazily follow the continuation tokens returned by the service and stop when all containers have been returned or *num_results* is reached.

If *num_results* is specified and the account has more than that number of containers, the generator will have a populated *next_marker* field once it finishes. This marker can be used to create a new generator if more results are desired.

Parameters

- **prefix** (*str*) – Filters the results to return only containers whose names begin with the specified prefix.
- **num_results** (*int*) – Specifies the maximum number of containers to return. A single list request may return up to 1000 containers and potentially a continuation token which should be followed to get additional results.
- **include_metadata** (*bool*) – Specifies that container metadata be returned in the response.
- **marker** (*str*) – An opaque continuation token. This value can be retrieved from the *next_marker* field of a previous generator object if *num_results* was specified and that generator has finished enumerating results. If specified, this generator will begin returning results from the point where the previous generator stopped.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

make_blob_url (*container_name, blob_name, protocol=None, sas_token=None, snapshot=None*)

Creates the url to access a blob.

Parameters

- **container_name** (*str*) – Name of container.
- **blob_name** (*str*) – Name of blob.
- **protocol** (*str*) – Protocol to use: 'http' or 'https'. If not specified, uses the protocol specified when *BaseBlobService* was initialized.
- **sas_token** (*str*) – Shared access signature token created with *generate_shared_access_signature*.
- **snapshot** (*str*) – An string value that uniquely identifies the snapshot. The value of this query parameter indicates the snapshot version.

Returns blob access URL.

Return type `str`

release_blob_lease (*container_name*, *blob_name*, *lease_id*, *if_modified_since=None*, *if_unmodified_since=None*, *if_match=None*, *if_none_match=None*, *timeout=None*)

Releases the lease. The lease may be released if the lease ID specified on the request matches that associated with the blob. Releasing the lease allows another client to immediately acquire the lease for the blob as soon as the release is complete.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **lease_id** (*str*) – Lease ID for active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

release_container_lease (*container_name*, *lease_id*, *if_modified_since=None*, *if_unmodified_since=None*, *timeout=None*)

Release the lease. The lease may be released if the *lease_id* specified matches that associated with the container. Releasing the lease allows another client to immediately acquire the lease for the container as soon as the release is complete.

Parameters

- **container_name** (*str*) – Name of existing container.
- **lease_id** (*str*) – Lease ID for active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.

- **timeout** (*int*) – The timeout parameter is expressed in seconds.

renew_blob_lease (*container_name*, *blob_name*, *lease_id*, *if_modified_since=None*, *if_unmodified_since=None*, *if_match=None*, *if_none_match=None*, *timeout=None*)

Renews the lease. The lease can be renewed if the lease ID specified on the request matches that associated with the blob. Note that the lease may be renewed even if it has expired as long as the blob has not been modified or leased again since the expiration of that lease. When you renew a lease, the lease duration clock resets.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **lease_id** (*str*) – Lease ID for active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns the lease ID of the renewed lease.

Returns str

renew_container_lease (*container_name*, *lease_id*, *if_modified_since=None*, *if_unmodified_since=None*, *timeout=None*)

Renews the lease. The lease can be renewed if the lease ID specified matches that associated with the container. Note that the lease may be renewed even if it has expired as long as the container has not been leased again since the expiration of that lease. When you renew a lease, the lease duration clock resets.

Parameters

- **container_name** (*str*) – Name of existing container.
- **lease_id** (*str*) – Lease ID for active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.

- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns the lease ID of the renewed lease.

Returns str

```
set_blob_metadata (container_name, blob_name, metadata=None, lease_id=None,
                   if_modified_since=None, if_unmodified_since=None, if_match=None,
                   if_none_match=None, timeout=None)
```

Sets user-defined metadata for the specified blob as one or more name-value pairs.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **metadata** (*a dict mapping str to str*) – Dict containing name and value pairs. Each call to this operation replaces all existing metadata attached to the blob. To remove all metadata from the blob, call this operation with no metadata headers.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns ETag and last modified properties for the updated Blob

Return type *ResourceProperties*

```
set_blob_properties (container_name, blob_name, content_settings=None, lease_id=None,
                     if_modified_since=None, if_unmodified_since=None, if_match=None,
                     if_none_match=None, timeout=None)
```

Sets system properties on the blob. If one property is set for the content_settings, all properties will be overridden.

Parameters

- **container_name** (*str*) – Name of existing container.

- **blob_name** (*str*) – Name of existing blob.
- **content_settings** (*ContentSettings*) – ContentSettings object used to set blob properties.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns ETag and last modified properties for the updated Blob

Return type *ResourceProperties*

set_blob_service_properties (*logging=None, hour_metrics=None, minute_metrics=None, cors=None, target_version=None, timeout=None*)

Sets the properties of a storage account’s Blob service, including Azure Storage Analytics. If an element (ex Logging) is left as None, the existing settings on the service for that functionality are preserved.

Parameters

- **logging** (*Logging*) – Groups the Azure Analytics Logging settings.
- **hour_metrics** (*Metrics*) – The hour metrics settings provide a summary of request statistics grouped by API in hourly aggregates for blobs.
- **minute_metrics** (*Metrics*) – The minute metrics settings provide request statistics for each minute for blobs.
- **cors** (list of *CorsRule*) – You can include up to five CorsRule elements in the list. If an empty list is specified, all CORS rules will be deleted, and CORS will be disabled for the service.
- **target_version** (*string*) – Indicates the default version to use for requests if an incoming request’s version is not specified.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

set_container_acl (*container_name, signed_identifiers=None, public_access=None, lease_id=None, if_modified_since=None, if_unmodified_since=None, timeout=None*)

Sets the permissions for the specified container or stored access policies that may be used with Shared Access Signatures. The permissions indicate whether blobs in a container may be accessed publicly.

Parameters

- **container_name** (*str*) – Name of existing container.
- **signed_identifiers** (dict of str to *AccessPolicy*) – A dictionary of access policies to associate with the container. The dictionary may contain up to 5 elements. An empty dictionary will clear the access policies set on the service.
- **public_access** (One of the values listed in the *PublicAccess* enum.) – Possible values include: container, blob.
- **lease_id** (*str*) – If specified, `set_container_acl` only succeeds if the container’s lease is active and matches this ID.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns ETag and last modified properties for the updated Container

Return type *ResourceProperties*

set_container_metadata (*container_name*, *metadata=None*, *lease_id=None*, *if_modified_since=None*, *timeout=None*)

Sets one or more user-defined name-value pairs for the specified container. Each call to this operation replaces all existing metadata attached to the container. To remove all metadata from the container, call this operation with no metadata dict.

Parameters

- **container_name** (*str*) – Name of existing container.
- **metadata** (*a dict mapping str to str*) – A dict containing name-value pairs to associate with the container as metadata. Example: {‘category’:‘test’}
- **lease_id** (*str*) – If specified, `set_container_metadata` only succeeds if the container’s lease is active and matches this ID.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns ETag and last modified properties for the updated Container

Return type *ResourceProperties*

snapshot_blob (*container_name*, *blob_name*, *metadata=None*, *if_modified_since=None*, *if_unmodified_since=None*, *if_match=None*, *if_none_match=None*, *lease_id=None*, *timeout=None*)

Creates a read-only snapshot of a blob.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **metadata** (*a dict mapping str to str*) – Specifies a user-defined name-value pair associated with the blob. If no name-value pairs are specified, the operation will copy

the base blob metadata to the snapshot. If one or more name-value pairs are specified, the snapshot is created with the specified metadata, and metadata is not copied from the base blob.

- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns snapshot properties

Return type *Blob*

azure.storage.blob.blockblobservice module

```
class azure.storage.blob.blockblobservice.BlockBlobService (account_name=None,
                                                            account_key=None,
                                                            sas_token=None,
                                                            is_emulated=False,
                                                            protocol='https', end-
                                                            point_suffix='core.windows.net',
                                                            custom_domain=None,
                                                            request_session=None,
                                                            connection_string=None,
                                                            socket_timeout=None)
```

Bases: *azure.storage.blob.baseblobservice.BaseBlobService*

Block blobs let you upload large blobs efficiently. Block blobs are comprised of blocks, each of which is identified by a block ID. You create or modify a block blob by writing a set of blocks and committing them by their block IDs. Each block can be a different size, up to a maximum of 4 MB, and a block blob can include up to 50,000 blocks. The maximum size of a block blob is therefore slightly more than 195 GB (4 MB X 50,000 blocks). If you are writing a block blob that is no more than 64 MB in size, you can upload it in its entirety with a single write operation; see `create_blob_from_bytes`.

Variables

- **MAX_SINGLE_PUT_SIZE** (*int*) – The largest size upload supported in a single put call. This is used by the `create_blob_from_*` methods if the content length is known and is less than this value.

- **MAX_BLOCK_SIZE** (*int*) – The size of the blocks put by `create_blob_from_*` methods if the content length is unknown or is larger than `MAX_SINGLE_PUT_SIZE`. Smaller blocks may be put. The maximum block size the service supports is 100MB.
- **MIN_LARGE_BLOCK_UPLOAD_THRESHOLD** (*int*) – The minimum block size at which the memory-optimized, block upload algorithm is considered. This algorithm is only applicable to the `create_blob_from_file` and `create_blob_from_stream` methods and will prevent the full buffering of blocks. In addition to the block size, ContentMD5 validation and Encryption must be disabled as these options require the blocks to be buffered.

Parameters

- **account_name** (*str*) – The storage account name. This is used to authenticate requests signed with an account key and to construct the storage endpoint. It is required unless a connection string is given, or if a custom domain is used with anonymous authentication.
- **account_key** (*str*) – The storage account key. This is used for shared key authentication. If neither account key or sas token is specified, anonymous access will be used.
- **sas_token** (*str*) – A shared access signature token to use to authenticate requests instead of the account key. If account key and sas token are both specified, account key will be used to sign. If neither are specified, anonymous access will be used.
- **is_emulated** (*bool*) – Whether to use the emulator. Defaults to False. If specified, will override all other parameters besides connection string and request session.
- **protocol** (*str*) – The protocol to use for requests. Defaults to https.
- **endpoint_suffix** (*str*) – The host base component of the url, minus the account name. Defaults to Azure (core.windows.net). Override this to use the China cloud (core.chinacloudapi.cn).
- **custom_domain** (*str*) – The custom domain to use. This can be set in the Azure Portal. For example, 'www.mydomain.com'.
- **request_session** (*requests.Session*) – The session object to use for http requests.
- **connection_string** (*str*) – If specified, this will override all other parameters besides request session. See <http://azure.microsoft.com/en-us/documentation/articles/storage-configure-connection-string/> for the connection string format.
- **socket_timeout** (*int*) – If specified, this will override the default socket timeout. The timeout specified is in seconds. See `DEFAULT_SOCKET_TIMEOUT` in `_constants.py` for the default value.

MAX_BLOCK_SIZE = 4194304

MAX_SINGLE_PUT_SIZE = 67108864

MIN_LARGE_BLOCK_UPLOAD_THRESHOLD = 4194305

create_blob_from_bytes (*container_name, blob_name, blob, index=0, count=None, content_settings=None, metadata=None, validate_content=False, progress_callback=None, max_connections=2, lease_id=None, if_modified_since=None, if_unmodified_since=None, if_match=None, if_none_match=None, timeout=None*)

Creates a new blob from an array of bytes, or updates the content of an existing blob, with automatic chunking and progress notifications.

Parameters

- **container_name** (*str*) – Name of existing container.

- **blob_name** (*str*) – Name of blob to create or update.
- **blob** (*bytes*) – Content of blob as an array of bytes.
- **index** (*int*) – Start index in the array of bytes.
- **count** (*int*) – Number of bytes to upload. Set to None or negative value to upload all bytes starting from index.
- **content_settings** (*ContentSettings*) – ContentSettings object used to set blob properties.
- **metadata** (*a dict mapping str to str*) – Name-value pairs associated with the blob as metadata.
- **validate_content** (*bool*) – If true, calculates an MD5 hash for each chunk of the blob. The storage service checks the hash of the content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the blob.
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature function(current, total) where current is the number of bytes transferred so far, and total is the size of the blob, or None if the total size is unknown.
- **max_connections** (*int*) – Maximum number of parallel connections to use when the blob size exceeds 64MB.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

Returns ETag and last modified properties for the Block Blob

Return type *ResourceProperties*

```
create_blob_from_path(container_name, blob_name, file_path, content_settings=None, metadata=None, validate_content=False, progress_callback=None, max_connections=2, lease_id=None, if_modified_since=None, if_unmodified_since=None, if_match=None, if_none_match=None, timeout=None)
```

Creates a new blob from a file path, or updates the content of an existing blob, with automatic chunking and progress notifications.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of blob to create or update.
- **file_path** (*str*) – Path of the file to upload as the blob content.
- **content_settings** (*ContentSettings*) – *ContentSettings* object used to set blob properties.
- **metadata** (*a dict mapping str to str*) – Name-value pairs associated with the blob as metadata.
- **validate_content** (*bool*) – If true, calculates an MD5 hash for each chunk of the blob. The storage service checks the hash of the content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the blob.
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature *function(current, total)* where *current* is the number of bytes transferred so far, and *total* is the size of the blob, or *None* if the total size is unknown.
- **max_connections** (*int*) – Maximum number of parallel connections to use when the blob size exceeds 64MB.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A *DateTime* value. Azure expects the date value passed in to be UTC. If *timezone* is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without *timezone* info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A *DateTime* value. Azure expects the date value passed in to be UTC. If *timezone* is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without *timezone* info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

Returns ETag and last modified properties for the Block Blob

Return type *ResourceProperties*


```
create_blob_from_stream(container_name, blob_name, stream, count=None, content_settings=None, metadata=None, validate_content=False, progress_callback=None, max_connections=2, lease_id=None, if_modified_since=None, if_unmodified_since=None, if_match=None, if_none_match=None, timeout=None, use_byte_buffer=False)
```

Creates a new blob from a file/stream, or updates the content of an existing blob, with automatic chunking and progress notifications.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of blob to create or update.
- **stream** (*io.IOBase*) – Opened file/stream to upload as the blob content.
- **count** (*int*) – Number of bytes to read from the stream. This is optional, but should be supplied for optimal performance.
- **content_settings** (*ContentSettings*) – ContentSettings object used to set blob properties.
- **metadata** (*a dict mapping str to str*) – Name-value pairs associated with the blob as metadata.
- **validate_content** (*bool*) – If true, calculates an MD5 hash for each chunk of the blob. The storage service checks the hash of the content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the blob.
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature function(current, total) where current is the number of bytes transferred so far, and total is the size of the blob, or None if the total size is unknown.
- **max_connections** (*int*) – Maximum number of parallel connections to use when the blob size exceeds 64MB. Note that parallel upload requires the stream to be seekable.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.

- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.
- **use_byte_buffer** (*bool*) – If True, this will force usage of the original full block buffering upload path. By default, this value is False and will employ a memory-efficient, streaming upload algorithm under the following conditions: The provided stream is seekable, 'require_encryption' is False, and MAX_BLOCK_SIZE >= MIN_LARGE_BLOCK_UPLOAD_THRESHOLD. One should consider the drawbacks of using this approach. In order to achieve memory-efficiency, a IOBase stream or file-like object is segmented into logical blocks using a SubStream wrapper. In order to read the correct data, each SubStream must acquire a lock so that it can safely seek to the right position on the shared, underlying stream. If max_connections > 1, the concurrency will result in a considerable amount of seeking on the underlying stream. For the most common inputs such as a file-like stream object, seeking is an inexpensive operation and this is not much of a concern. However, for other variants of streams this may not be the case. The trade-off for memory-efficiency must be weighed against the cost of seeking with your input stream. The SubStream class will attempt to buffer up to 4 MB internally to reduce the amount of seek and read calls to the underlying stream. This is particularly beneficial when uploading larger blocks.

Returns ETag and last modified properties for the Block Blob

Return type *ResourceProperties*

create_blob_from_text (*container_name, blob_name, text, encoding='utf-8', content_settings=None, metadata=None, validate_content=False, progress_callback=None, max_connections=2, lease_id=None, if_modified_since=None, if_unmodified_since=None, if_match=None, if_none_match=None, timeout=None*)

Creates a new blob from str/unicode, or updates the content of an existing blob, with automatic chunking and progress notifications.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of blob to create or update.
- **text** (*str*) – Text to upload to the blob.
- **encoding** (*str*) – Python encoding to use to convert the text to bytes.
- **content_settings** (*ContentSettings*) – ContentSettings object used to set blob properties.
- **metadata** (*a dict mapping str to str*) – Name-value pairs associated with the blob as metadata.
- **validate_content** (*bool*) – If true, calculates an MD5 hash for each chunk of the blob. The storage service checks the hash of the content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the blob.
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature function(current, total) where current is the number of bytes transferred so far, and total is the size of the blob, or None if the total size is unknown.
- **max_connections** (*int*) – Maximum number of parallel connections to use when the blob size exceeds 64MB.

- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

Returns ETag and last modified properties for the Block Blob

Return type *ResourceProperties*

get_block_list (*container_name*, *blob_name*, *snapshot=None*, *block_list_type=None*, *lease_id=None*, *timeout=None*)

Retrieves the list of blocks that have been uploaded as part of a block blob. There are two block lists maintained for a blob:

Committed Block List: The list of blocks that have been successfully committed to a given blob with Put Block List.

Uncommitted Block List: The list of blocks that have been uploaded for a blob using Put Block, but that have not yet been committed. These blocks are stored in Azure in association with a blob, but do not yet form part of the blob.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **snapshot** (*str*) – Datetime to determine the time to retrieve the blocks.
- **block_list_type** (*str*) – Specifies whether to return the list of committed blocks, the list of uncommitted blocks, or both lists together. Valid values are: committed, uncommitted, or all.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns list committed and/or uncommitted blocks for Block Blob

Return type *BlobBlockList*

put_block (*container_name*, *blob_name*, *block*, *block_id*, *validate_content=False*, *lease_id=None*, *timeout=None*)

Creates a new block to be committed as part of a blob.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **block** (*io.IOBase* or *bytes* Content of the block.) – Content of the block.
- **block_id** (*str*) – A valid Base64 string value that identifies the block. Prior to encoding, the string must be less than or equal to 64 bytes in size. For a given blob, the length of the value specified for the blockid parameter must be the same size for each block. Note that the Base64 string must be URL-encoded.
- **validate_content** (*bool*) – If true, calculates an MD5 hash of the block content. The storage service checks the hash of the content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the blob.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

put_block_list (*container_name*, *blob_name*, *block_list*, *content_settings=None*, *metadata=None*, *validate_content=False*, *lease_id=None*, *if_modified_since=None*, *if_unmodified_since=None*, *if_match=None*, *if_none_match=None*, *timeout=None*)

Writes a blob by specifying the list of block IDs that make up the blob. In order to be written as part of a blob, a block must have been successfully written to the server in a prior Put Block operation.

You can call Put Block List to update a blob by uploading only those blocks that have changed, then committing the new and existing blocks together. You can do this by specifying whether to commit a block from the committed block list or from the uncommitted block list, or to commit the most recently uploaded version of the block, whichever list it may belong to.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **block_list** (list of *BlobBlock*) – A list of `BlobBlock` containing the block ids and block state.
- **content_settings** (*ContentSettings*) – `ContentSettings` object used to set properties on the blob.
- **metadata** (*a dict mapping str to str*) – Name-value pairs associated with the blob as metadata.
- **validate_content** (*bool*) – If true, calculates an MD5 hash of the block list content. The storage service checks the hash of the block list content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this check is associated with the block list content, and not with the content of the blob itself.
- **lease_id** (*str*) – Required if the blob has an active lease.

- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns ETag and last modified properties for the updated Block Blob

Return type *ResourceProperties*

set_standard_blob_tier (*container_name, blob_name, standard_blob_tier, timeout=None*)

Sets the block blob tiers on the blob. This API is only supported for block blobs on standard storage accounts.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of blob to update.
- **standard_blob_tier** (*StandardBlobTier*) – A standard blob tier value to set the blob to. For this version of the library, this is only applicable to block blobs on standard storage accounts.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

azure.storage.blob.models module

class `azure.storage.blob.models.AppendBlockProperties`

Bases: *azure.storage.blob.models.ResourceProperties*

Response for an append block request.

Variables

- **append_offset** (*int*) – Position to start next append.
- **committed_block_count** (*int*) – Number of committed append blocks.

class `azure.storage.blob.models.Blob` (*name=None, snapshot=None, content=None, props=None, metadata=None*)

Bases: object

Blob class.

Variables

- **name** (*str*) – Name of blob.
- **snapshot** (*str*) – A DateTime value that uniquely identifies the snapshot. The value of this header indicates the snapshot version, and may be used in subsequent requests to access the snapshot.
- **content** (*str or bytes*) – Blob content.
- **properties** (*BlobProperties*) – Stores all the system properties for the blob.
- **metadata** – Name-value pairs associated with the blob as metadata.

class `azure.storage.blob.models.BlobBlock` (*id=None, state='Latest'*)

Bases: `object`

BlockBlob Block class.

Variables

- **id** (*str*) – Block id.
- **state** (*str*) – Block state. Possible value: `committeduncommitted`
- **size** (*int*) – Block size in bytes.

class `azure.storage.blob.models.BlobBlockList`

Bases: `object`

Blob Block List class.

Variables

- **committed_blocks** (list of *BlobBlock*) – List of committed blocks.
- **uncommitted_blocks** (list of *BlobBlock*) – List of uncommitted blocks.

class `azure.storage.blob.models.BlobBlockState`

Bases: `object`

Block blob block types.

Committed = 'Committed'

Committed blocks.

Latest = 'Latest'

Latest blocks.

Uncommitted = 'Uncommitted'

Uncommitted blocks.

class `azure.storage.blob.models.BlobPermissions` (*read=False, add=False, create=False, write=False, delete=False, _str=None*)

Bases: `object`

BlobPermissions class to be used with `generate_blob_shared_access_signature()` API.

Variables

- **BlobPermissions.ADD** (*BlobPermissions*) – Add a block to an append blob.
- **BlobPermissions.CREATE** (*BlobPermissions*) – Write a new blob, snapshot a blob, or copy a blob to a new blob.
- **BlobPermissions.DELETE** (*BlobPermissions*) – Delete the blob.
- **BlobPermissions.READ** (*BlobPermissions*) – Read the content, properties, metadata and block list. Use the blob as the source of a copy operation.

- **BlobPermissions.WRITE** (`BlobPermissions`) – Create or write content, properties, metadata, or block list. Snapshot or lease the blob. Resize the blob (page blob only). Use the blob as the destination of a copy operation within the same account.

Parameters

- **read** (*bool*) – Read the content, properties, metadata and block list. Use the blob as the source of a copy operation.
- **add** (*bool*) – Add a block to an append blob.
- **create** (*bool*) – Write a new blob, snapshot a blob, or copy a blob to a new blob.
- **write** (*bool*) – Create or write content, properties, metadata, or block list. Snapshot or lease the blob. Resize the blob (page blob only). Use the blob as the destination of a copy operation within the same account.
- **delete** (*bool*) – Delete the blob.
- **_str** (*str*) – A string representing the permissions.

ADD = <azure.storage.blob.models.BlobPermissions object>

CREATE = <azure.storage.blob.models.BlobPermissions object>

DELETE = <azure.storage.blob.models.BlobPermissions object>

READ = <azure.storage.blob.models.BlobPermissions object>

WRITE = <azure.storage.blob.models.BlobPermissions object>

class azure.storage.blob.models.**BlobPrefix**

Bases: object

BlobPrefix objects may potentially returned in the blob list when `list_blobs()` is used with a delimiter. Prefixes can be thought of as virtual blob directories.

Variables **name** (*str*) – The name of the blob prefix.

class azure.storage.blob.models.**BlobProperties**

Bases: object

Blob Properties

Variables

- **blob_type** (*str*) – String indicating this blob’s type.
- **last_modified** (*datetime*) – A datetime object representing the last time the blob was modified.
- **etag** (*str*) – The ETag contains a value that you can use to perform operations conditionally.
- **content_length** (*int*) – The length of the content returned. If the entire blob was requested, the length of blob in bytes. If a subset of the blob was requested, the length of the returned subset.
- **content_range** (*str*) – Indicates the range of bytes returned in the event that the client requested a subset of the blob.
- **append_blob_committed_block_count** (*int*) – (For Append Blobs) Number of committed blocks in the blob.
- **page_blob_sequence_number** (*int*) – (For Page Blobs) Sequence number for page blob used for coordinating concurrent writes.

- **server_encrypted** (*bool*) – Set to true if the blob is encrypted on the server.
- **copy** (*CopyProperties*) – Stores all the copy properties for the blob.
- **content_settings** (*ContentSettings*) – Stores all the content settings for the blob.
- **lease** (*LeaseProperties*) – Stores all the lease information for the blob.

class `azure.storage.blob.models.BlockListType`

Bases: `object`

Specifies whether to return the list of committed blocks, the list of uncommitted blocks, or both lists together.

All = 'all'

Both committed and uncommitted blocks.

Committed = 'committed'

Committed blocks.

Uncommitted = 'uncommitted'

Uncommitted blocks.

class `azure.storage.blob.models.Container` (*name=None, props=None, metadata=None*)

Bases: `object`

Blob container class.

Variables

- **name** (*str*) – The name of the container.
- **metadata** (*dict mapping str to str*) – A dict containing name-value pairs associated with the container as metadata. This var is set to None unless the `include=metadata` param was included for the list containers operation. If this parameter was specified but the container has no metadata, metadata will be set to an empty dictionary.
- **properties** (*ContainerProperties*) – System properties for the container.

class `azure.storage.blob.models.ContainerPermissions` (*read=False, write=False, delete=False, list=False, _str=None*)

Bases: `object`

ContainerPermissions class to be used with `generate_container_shared_access_signature()` API and for the AccessPolicies used with `set_container_acl()`.

Variables

- **ContainerPermissions.DELETE** (*ContainerPermissions*) – Delete any blob in the container. Note: You cannot grant permissions to delete a container with a container SAS. Use an account SAS instead.
- **ContainerPermissions.LIST** (*ContainerPermissions*) – List blobs in the container.
- **ContainerPermissions.READ** (*ContainerPermissions*) – Read the content, properties, metadata or block list of any blob in the container. Use any blob in the container as the source of a copy operation.
- **ContainerPermissions.WRITE** (*ContainerPermissions*) – For any blob in the container, create or write content, properties, metadata, or block list. Snapshot or lease the blob. Resize the blob (page blob only). Use the blob as the destination of a copy operation within the same account. Note: You cannot grant permissions to read or write

container properties or metadata, nor to lease a container, with a container SAS. Use an account SAS instead.

Parameters

- **read** (*bool*) – Read the content, properties, metadata or block list of any blob in the container. Use any blob in the container as the source of a copy operation.
- **write** (*bool*) – For any blob in the container, create or write content, properties, metadata, or block list. Snapshot or lease the blob. Resize the blob (page blob only). Use the blob as the destination of a copy operation within the same account. Note: You cannot grant permissions to read or write container properties or metadata, nor to lease a container, with a container SAS. Use an account SAS instead.
- **delete** (*bool*) – Delete any blob in the container. Note: You cannot grant permissions to delete a container with a container SAS. Use an account SAS instead.
- **list** (*bool*) – List blobs in the container.
- **_str** (*str*) – A string representing the permissions.

DELETE = <azure.storage.blob.models.ContainerPermissions object>

LIST = <azure.storage.blob.models.ContainerPermissions object>

READ = <azure.storage.blob.models.ContainerPermissions object>

WRITE = <azure.storage.blob.models.ContainerPermissions object>

class azure.storage.blob.models.**ContainerProperties**

Bases: object

Blob container’s properties class.

Variables

- **last_modified** (*datetime*) – A datetime object representing the last time the container was modified.
- **etag** (*str*) – The ETag contains a value that you can use to perform operations conditionally.
- **lease** (*LeaseProperties*) – Stores all the lease information for the container.

class azure.storage.blob.models.**ContentSettings** (*content_type=None, content_encoding=None, content_language=None, content_disposition=None, cache_control=None, content_md5=None*)

Bases: object

Used to store the content settings of a blob.

Variables

- **content_type** (*str*) – The content type specified for the blob. If no content type was specified, the default content type is application/octet-stream.
- **content_encoding** (*str*) – If the content_encoding has previously been set for the blob, that value is stored.
- **content_language** (*str*) – If the content_language has previously been set for the blob, that value is stored.

- **content_disposition** (*str*) – content_disposition conveys additional information about how to process the response payload, and also can be used to attach additional meta-data. If content_disposition has previously been set for the blob, that value is stored.
- **cache_control** (*str*) – If the cache_control has previously been set for the blob, that value is stored.
- **content_md5** (*str*) – If the content_md5 has been set for the blob, this response header is stored so that the client can check for message content integrity.

class azure.storage.blob.models.**CopyProperties**

Bases: object

Blob Copy Properties.

Variables

- **id** (*str*) – String identifier for the last attempted Copy Blob operation where this blob was the destination blob. This header does not appear if this blob has never been the destination in a Copy Blob operation, or if this blob has been modified after a concluded Copy Blob operation using Set Blob Properties, Put Blob, or Put Block List.
- **source** (*str*) – URL up to 2 KB in length that specifies the source blob used in the last attempted Copy Blob operation where this blob was the destination blob. This header does not appear if this blob has never been the destination in a Copy Blob operation, or if this blob has been modified after a concluded Copy Blob operation using Set Blob Properties, Put Blob, or Put Block List.
- **status** (*str*) –

State of the copy operation identified by Copy ID, with these values:

success: Copy completed successfully.

pending: Copy is in progress. Check copy_status_description if intermittent, non-fatal errors impede copy progress but don't cause failure.

aborted: Copy was ended by Abort Copy Blob.

failed: Copy failed. See copy_status_description for failure details.

- **progress** (*str*) – Contains the number of bytes copied and the total bytes in the source in the last attempted Copy Blob operation where this blob was the destination blob. Can show between 0 and Content-Length bytes copied.
- **completion_time** (*datetime*) – Conclusion time of the last attempted Copy Blob operation where this blob was the destination blob. This value can specify the time of a completed, aborted, or failed copy attempt.
- **status_description** (*str*) – only appears when x-ms-copy-status is failed or pending. Describes cause of fatal or non-fatal copy operation failure.

class azure.storage.blob.models.**DeleteSnapshot**

Bases: object

Required if the blob has associated snapshots. Specifies how to handle the snapshots.

Include = 'include'

Delete the base blob and all of its snapshots.

Only = 'only'

Delete only the blob's snapshots and not the blob itself.

class `azure.storage.blob.models.Include` (*snapshots=False, metadata=False, uncommitted_blobs=False, copy=False, _str=None*)

Bases: `object`

Specifies the datasets to include in the blob list response.

Variables

- ***Include.COPY*** (*Include*) – Specifies that metadata related to any current or previous Copy Blob operation should be included in the response.
- ***Include.METADATA*** (*Include*) – Specifies that metadata be returned in the response.
- ***Include.SNAPSHOTS*** (*Include*) – Specifies that snapshots should be included in the enumeration.
- ***Include.UNCOMMITTED_BLOBS*** (*Include*) – Specifies that blobs for which blocks have been uploaded, but which have not been committed using Put Block List, be included in the response.

Parameters

- **`snapshots`** (*bool*) – Specifies that snapshots should be included in the enumeration.
- **`metadata`** (*bool*) – Specifies that metadata be returned in the response.
- **`uncommitted_blobs`** (*bool*) – Specifies that blobs for which blocks have been uploaded, but which have not been committed using Put Block List, be included in the response.
- **`copy`** (*bool*) – Specifies that metadata related to any current or previous Copy Blob operation should be included in the response.
- **`_str`** (*str*) – A string representing the includes.

COPY = <`azure.storage.blob.models.Include` object>

METADATA = <`azure.storage.blob.models.Include` object>

SNAPSHOTS = <`azure.storage.blob.models.Include` object>

UNCOMMITTED_BLOBS = <`azure.storage.blob.models.Include` object>

class `azure.storage.blob.models.LeaseProperties`

Bases: `object`

Blob Lease Properties.

Variables

- **`status`** (*str*) – The lease status of the blob. Possible values: `locked`/`unlocked`
- **`state`** (*str*) – Lease state of the blob. Possible values: `available`/`leased`/`expired`/`breaking`/`broken`
- **`duration`** (*str*) – When a blob is leased, specifies whether the lease is of infinite or fixed duration.

class `azure.storage.blob.models.PageBlobProperties`

Bases: `azure.storage.blob.models.ResourceProperties`

Response for a page request.

Variables **`sequence_number`** (*int*) – Identifier for page blobs to help handle concurrent writes.

class `azure.storage.blob.models.PageRange` (*start=None, end=None, is_cleared=False*)
 Bases: `object`

Page Range for page blob.

Variables

- **start** (*int*) – Start of page range in bytes.
- **end** (*int*) – End of page range in bytes.
- **is_cleared** (*bool*) – Indicates if a page range is cleared or not. Only applicable for `get_page_range_diff` API.

class `azure.storage.blob.models.PremiumPageBlobTier`
 Bases: `object`

Specifies the page blob tier to set the blob to. This is only applicable to page blobs on premium storage accounts. Please take a look at <https://docs.microsoft.com/en-us/azure/storage/storage-premium-storage#scalability-and-performance-targets> for detailed information on the corresponding IOPS and throughput per `PageBlobTier`.

P10 = 'P10'
 P10 Tier

P20 = 'P20'
 P20 Tier

P30 = 'P30'
 P30 Tier

P4 = 'P4'
 P4 Tier

P40 = 'P40'
 P40 Tier

P50 = 'P50'
 P50 Tier

P6 = 'P6'
 P6 Tier

P60 = 'P60'
 P60 Tier

class `azure.storage.blob.models.PublicAccess`
 Bases: `object`

Specifies whether data in the container may be accessed publicly and the level of access.

Blob = 'blob'

Specifies public read access for blobs. Blob data within this container can be read via anonymous request, but container data is not available. Clients cannot enumerate blobs within the container via anonymous request.

Container = 'container'

Specifies full public read access for container and blob data. Clients can enumerate blobs within the container via anonymous request, but cannot enumerate containers within the storage account.

OFF = 'off'

Specifies that there is no public read access for both the container and blobs within the container. Clients cannot enumerate the containers within the storage account as well as the blobs within the container.

class `azure.storage.blob.models.ResourceProperties`

Bases: `object`

Base response for a resource request.

Variables

- **etag** (*str*) – Opaque etag value that can be used to check if resource has been modified.
- **last_modified** (*datetime*) – Datetime for last time resource was modified.

class `azure.storage.blob.models.SequenceNumberAction`

Bases: `object`

Sequence number actions.

Increment = 'increment'

Increments the value of the sequence number by 1. If specifying this option, do not include the x-ms-blob-sequence-number header.

Max = 'max'

Sets the sequence number to be the higher of the value included with the request and the value currently stored for the blob.

Update = 'update'

Sets the sequence number to the value included with the request.

class `azure.storage.blob.models.StandardBlobTier`

Bases: `object`

Specifies the blob tier to set the blob to. This is only applicable for block blobs on standard storage accounts.

Archive = 'Archive'

Archive

Cool = 'Cool'

Cool

Hot = 'Hot'

Hot

azure.storage.blob.pageblob service module

class `azure.storage.blob.pageblob service. PageBlobService` (*account_name=None, account_key=None, sas_token=None, is_emulated=False, protocol='https', endpoint_suffix='core.windows.net', custom_domain=None, request_session=None, connection_string=None, socket_timeout=None*)

Bases: `azure.storage.blob.baseblob service. BaseBlobService`

Page blobs are a collection of 512-byte pages optimized for random read and write operations. To create a page blob, you initialize the page blob and specify the maximum size the page blob will grow. To add or update the contents of a page blob, you write a page or pages by specifying an offset and a range that align to 512-byte page boundaries. A write to a page blob can overwrite just one page, some pages, or up to 4 MB of the page blob. Writes to page blobs happen in-place and are immediately committed to the blob. The maximum size for a page blob is 1 TB.

Variables `MAX_PAGE_SIZE` (*int*) – The size of the pages put by `create_blob_from_*` methods. Smaller pages may be put if there is less data provided. The maximum page size the service supports is 4MB.

Parameters

- **account_name** (*str*) – The storage account name. This is used to authenticate requests signed with an account key and to construct the storage endpoint. It is required unless a connection string is given, or if a custom domain is used with anonymous authentication.
- **account_key** (*str*) – The storage account key. This is used for shared key authentication. If neither account key or sas token is specified, anonymous access will be used.
- **sas_token** (*str*) – A shared access signature token to use to authenticate requests instead of the account key. If account key and sas token are both specified, account key will be used to sign. If neither are specified, anonymous access will be used.
- **is_emulated** (*bool*) – Whether to use the emulator. Defaults to False. If specified, will override all other parameters besides connection string and request session.
- **protocol** (*str*) – The protocol to use for requests. Defaults to https.
- **endpoint_suffix** (*str*) – The host base component of the url, minus the account name. Defaults to Azure (core.windows.net). Override this to use the China cloud (core.chinacloudapi.cn).
- **custom_domain** (*str*) – The custom domain to use. This can be set in the Azure Portal. For example, 'www.mydomain.com'.
- **request_session** (*requests.Session*) – The session object to use for http requests.
- **connection_string** (*str*) – If specified, this will override all other parameters besides request session. See <http://azure.microsoft.com/en-us/documentation/articles/storage-configure-connection-string/> for the connection string format.
- **socket_timeout** (*int*) – If specified, this will override the default socket timeout. The timeout specified is in seconds. See `DEFAULT_SOCKET_TIMEOUT` in `_constants.py` for the default value.

MAX_PAGE_SIZE = 4194304

```
clear_page (container_name, blob_name, start_range, end_range, lease_id=None,
            if_sequence_number_lte=None, if_sequence_number_lt=None,
            if_sequence_number_eq=None, if_modified_since=None, if_unmodified_since=None,
            if_match=None, if_none_match=None, timeout=None)
```

Clears a range of pages.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **start_range** (*int*) – Start of byte range to use for writing to a section of the blob. Pages must be aligned with 512-byte boundaries, the start offset must be a modulus of 512 and the end offset must be a modulus of 512-1. Examples of valid byte ranges are 0-511, 512-1023, etc.
- **end_range** (*int*) – End of byte range to use for writing to a section of the blob. Pages must be aligned with 512-byte boundaries, the start offset must be a modulus of 512 and the end offset must be a modulus of 512-1. Examples of valid byte ranges are 0-511, 512-1023, etc.

- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_sequence_number_lte** (*int*) – If the blob’s sequence number is less than or equal to the specified value, the request proceeds; otherwise it fails.
- **if_sequence_number_lt** (*int*) – If the blob’s sequence number is less than the specified value, the request proceeds; otherwise it fails.
- **if_sequence_number_eq** (*int*) – If the blob’s sequence number is equal to the specified value, the request proceeds; otherwise it fails.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify an ETag value for this conditional header to write the page only if the blob’s ETag value matches the value specified. If the values do not match, the Blob service fails.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify an ETag value for this conditional header to write the page only if the blob’s ETag value does not match the value specified. If the values are identical, the Blob service fails.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns ETag and last modified properties for the updated Page Blob

Return type *ResourceProperties*

```
copy_blob (container_name, blob_name, copy_source, metadata=None,
            source_if_modified_since=None, source_if_unmodified_since=None,
            source_if_match=None, source_if_none_match=None, destination_if_modified_since=None,
            destination_if_unmodified_since=None, destination_if_match=None,
            destination_if_none_match=None, destination_lease_id=None,
            source_lease_id=None, timeout=None, premium_page_blob_tier=None)
```

Copies a blob asynchronously. This operation returns a copy operation properties object, including a copy ID you can use to check or abort the copy operation. The Blob service copies blobs on a best-effort basis.

The source blob for a copy operation must be a page blob. If the destination blob already exists, it must be of the same blob type as the source blob. Any existing destination blob will be overwritten. The destination blob cannot be modified while a copy operation is in progress.

When copying from a page blob, the Blob service creates a destination page blob of the source blob’s length, initially containing all zeroes. Then the source page ranges are enumerated, and non-empty ranges are copied.

If the tier on the source blob is larger than the tier being passed to this copy operation or if the size of the blob exceeds the tier being passed to this copy operation then the operation will fail.

You can call `get_blob_properties` on the destination blob to check the status of the copy operation. The final blob will be committed when the copy completes.

Parameters

- **container_name** (*str*) – Name of the destination container. The container must exist.

- **blob_name** (*str*) – Name of the destination blob. If the destination blob exists, it will be overwritten. Otherwise, it will be created.
- **copy_source** (*str*) – A URL of up to 2 KB in length that specifies an Azure file or blob. The value should be URL-encoded as it would appear in a request URI. If the source is in another account, the source must either be public or must be authenticated via a shared access signature. If the source is public, no authentication is required. Examples: <https://myaccount.blob.core.windows.net/mycontainer/myblob> <https://myaccount.blob.core.windows.net/mycontainer/myblob?snapshot=<DateTime>> <https://otheraccount.blob.core.windows.net/mycontainer/myblob?sastoken>
- **metadata** (*A dict mapping str to str.*) – Name-value pairs associated with the blob as metadata. If no name-value pairs are specified, the operation will copy the metadata from the source blob or file to the destination blob. If one or more name-value pairs are specified, the destination blob is created with the specified metadata, and metadata is not copied from the source blob or file.
- **source_if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this conditional header to copy the blob only if the source blob has been modified since the specified date/time.
- **source_if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this conditional header to copy the blob only if the source blob has not been modified since the specified date/time.
- **source_if_match** (*ETag*) – An ETag value, or the wildcard character (*). Specify this conditional header to copy the source blob only if its ETag matches the value specified. If the ETag values do not match, the Blob service returns status code 412 (Precondition Failed). This header cannot be specified if the source is an Azure File.
- **source_if_none_match** (*ETag*) – An ETag value, or the wildcard character (*). Specify this conditional header to copy the blob only if its ETag does not match the value specified. If the values are identical, the Blob service returns status code 412 (Precondition Failed). This header cannot be specified if the source is an Azure File.
- **destination_if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this conditional header to copy the blob only if the destination blob has been modified since the specified date/time. If the destination blob has not been modified, the Blob service returns status code 412 (Precondition Failed).
- **destination_if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this conditional header to copy the blob only if the destination blob has not been modified since the specified date/time. If the destination blob has been modified, the Blob service returns status code 412 (Precondition Failed).
- **destination_if_match** (*ETag*) – An ETag value, or the wildcard character (*). Specify an ETag value for this conditional header to copy the blob only if the specified ETag value matches the ETag value for an existing destination blob. If the ETag for the destination blob does not match the ETag specified for If-Match, the Blob service returns status code 412 (Precondition Failed).

- **destination_if_none_match** (*ETag*) – An ETag value, or the wildcard character (*). Specify an ETag value for this conditional header to copy the blob only if the specified ETag value does not match the ETag value for the destination blob. Specify the wildcard character (*) to perform the operation only if the destination blob does not exist. If the specified condition isn't met, the Blob service returns status code 412 (Precondition Failed).
- **destination_lease_id** (*str*) – The lease ID specified for this header must match the lease ID of the destination blob. If the request does not include the lease ID or it is not valid, the operation fails with status code 412 (Precondition Failed).
- **source_lease_id** (*str*) – Specify this to perform the Copy Blob operation only if the lease ID given matches the active lease ID of the source blob.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.
- **premium_page_blob_tier** (*PageBlobTier*) – A page blob tier value to set on the destination blob. The tier correlates to the size of the blob and number of allowed IOPS. This is only applicable to page blobs on premium storage accounts. If the tier on the source blob is larger than the tier being passed to this copy operation or if the size of the blob exceeds the tier being passed to this copy operation then the operation will fail.

Returns Copy operation properties such as status, source, and ID.

Return type *CopyProperties*

create_blob (*container_name*, *blob_name*, *content_length*, *content_settings=None*, *sequence_number=None*, *metadata=None*, *lease_id=None*, *if_modified_since=None*, *if_unmodified_since=None*, *if_match=None*, *if_none_match=None*, *timeout=None*, *premium_page_blob_tier=None*)

Creates a new Page Blob.

See `create_blob_from_*` for high level functions that handle the creation and upload of large blobs with automatic chunking and progress notifications.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of blob to create or update.
- **content_length** (*int*) – Required. This header specifies the maximum size for the page blob, up to 1 TB. The page blob size must be aligned to a 512-byte boundary.
- **content_settings** (*ContentSettings*) – *ContentSettings* object used to set properties on the blob.
- **sequence_number** (*int*) – The sequence number is a user-controlled value that you can use to track requests. The value of the sequence number must be between 0 and $2^{63} - 1$. The default value is 0.
- **metadata** (*a dict mapping str to str*) – Name-value pairs associated with the blob as metadata.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A *DateTime* value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.

- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.
- **premium_page_blob_tier** (*PremiumPageBlobTier*) – A page blob tier value to set the blob to. The tier correlates to the size of the blob and number of allowed IOPS. This is only applicable to page blobs on premium storage accounts.

Returns ETag and last modified properties for the new Page Blob

Return type *ResourceProperties*

create_blob_from_bytes (*container_name, blob_name, blob, index=0, count=None, content_settings=None, metadata=None, validate_content=False, progress_callback=None, max_connections=2, lease_id=None, if_modified_since=None, if_unmodified_since=None, if_match=None, if_none_match=None, timeout=None, premium_page_blob_tier=None*)

Creates a new blob from an array of bytes, or updates the content of an existing blob, with automatic chunking and progress notifications.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of blob to create or update.
- **blob** (*bytes*) – Content of blob as an array of bytes.
- **index** (*int*) – Start index in the byte array.
- **count** (*int*) – Number of bytes to upload. Set to None or negative value to upload all bytes starting from index.
- **content_settings** (*ContentSettings*) – ContentSettings object used to set blob properties.
- **metadata** (*a dict mapping str to str*) – Name-value pairs associated with the blob as metadata.
- **validate_content** (*bool*) – If true, calculates an MD5 hash for each page of the blob. The storage service checks the hash of the content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the blob.
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature function(current, total) where current is the number of bytes transferred so far, and total is the size of the blob, or None if the total size is unknown.

- **max_connections** (*int*) – Maximum number of parallel connections to use.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.
- **premium_page_blob_tier** – A page blob tier value to set the blob to. The tier correlates to the size of the blob and number of allowed IOPS. This is only applicable to page blobs on premium storage accounts.

Returns ETag and last modified properties for the Page Blob

Return type *ResourceProperties*

```
create_blob_from_path(container_name, blob_name, file_path, content_settings=None,  
                      metadata=None, validate_content=False, progress_callback=None,  
                      max_connections=2, lease_id=None, if_modified_since=None,  
                      if_unmodified_since=None, if_match=None, if_none_match=None,  
                      timeout=None, premium_page_blob_tier=None)
```

Creates a new blob from a file path, or updates the content of an existing blob, with automatic chunking and progress notifications.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of blob to create or update.
- **file_path** (*str*) – Path of the file to upload as the blob content.
- **content_settings** (*ContentSettings*) – ContentSettings object used to set blob properties.
- **metadata** (*a dict mapping str to str*) – Name-value pairs associated with the blob as metadata.
- **validate_content** (*bool*) – If true, calculates an MD5 hash for each page of the blob. The storage service checks the hash of the content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the blob.

- **progress_callback** (*callback function in format of `func(current, total)`*) – Callback for progress with signature `function(current, total)` where `current` is the number of bytes transferred so far, and `total` is the size of the blob, or `None` if the total size is unknown.
- **max_connections** (*int*) – Maximum number of parallel connections to use.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A `DateTime` value. Azure expects the date value passed in to be UTC. If `timezone` is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without `timezone` info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A `DateTime` value. Azure expects the date value passed in to be UTC. If `timezone` is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without `timezone` info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.
- **premium_page_blob_tier** – A page blob tier value to set the blob to. The tier correlates to the size of the blob and number of allowed IOPS. This is only applicable to page blobs on premium storage accounts.

Returns ETag and last modified properties for the Page Blob

Return type *ResourceProperties*

create_blob_from_stream(*container_name, blob_name, stream, count, content_settings=None, metadata=None, validate_content=False, progress_callback=None, max_connections=2, lease_id=None, if_modified_since=None, if_unmodified_since=None, if_match=None, if_none_match=None, timeout=None, premium_page_blob_tier=None*)

Creates a new blob from a file/stream, or updates the content of an existing blob, with automatic chunking and progress notifications.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of blob to create or update.
- **stream** (*io.IOBase*) – Opened file/stream to upload as the blob content.
- **count** (*int*) – Number of bytes to read from the stream. This is required, a page blob cannot be created if the count is unknown.
- **content_settings** (*ContentSettings*) – `ContentSettings` object used to set the blob properties.

- **metadata** (*a dict mapping str to str*) – Name-value pairs associated with the blob as metadata.
- **validate_content** (*bool*) – If true, calculates an MD5 hash for each page of the blob. The storage service checks the hash of the content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the blob.
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature function(current, total) where current is the number of bytes transfered so far, and total is the size of the blob, or None if the total size is unknown.
- **max_connections** (*int*) – Maximum number of parallel connections to use. Note that parallel upload requires the stream to be seekable.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.
- **premium_page_blob_tier** – A page blob tier value to set the blob to. The tier correlates to the size of the blob and number of allowed IOPS. This is only applicable to page blobs on premium storage accounts.

Returns ETag and last modified properties for the Page Blob

Return type *ResourceProperties*

get_page_ranges (*container_name, blob_name, snapshot=None, start_range=None, end_range=None, lease_id=None, if_modified_since=None, if_unmodified_since=None, if_match=None, if_none_match=None, timeout=None*)

Returns the list of valid page ranges for a Page Blob or snapshot of a page blob.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.

- **snapshot** (*str*) – The snapshot parameter is an opaque DateTime value that, when present, specifies the blob snapshot to retrieve information from.
- **start_range** (*int*) – Start of byte range to use for getting valid page ranges. If no end_range is given, all bytes after the start_range will be searched. Pages must be aligned with 512-byte boundaries, the start offset must be a modulus of 512 and the end offset must be a modulus of 512-1. Examples of valid byte ranges are 0-511, 512-, etc.
- **end_range** (*int*) – End of byte range to use for getting valid page ranges. If end_range is given, start_range must be provided. This range will return valid page ranges for from the offset start up to offset end. Pages must be aligned with 512-byte boundaries, the start offset must be a modulus of 512 and the end offset must be a modulus of 512-1. Examples of valid byte ranges are 0-511, 512-, etc.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns A list of valid Page Ranges for the Page Blob.

Return type list of *PageRange*

```
get_page_ranges_diff (container_name, blob_name, previous_snapshot, snapshot=None, start_range=None, end_range=None, lease_id=None, if_modified_since=None, if_unmodified_since=None, if_match=None, if_none_match=None, timeout=None)
```

The response will include only the pages that are different between either a recent snapshot or the current blob and a previous snapshot, including pages that were cleared.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **previous_snapshot** (*str*) – The snapshot parameter is an opaque DateTime value that specifies a previous blob snapshot to be compared against a more recent snapshot or the current blob.
- **snapshot** (*str*) – The snapshot parameter is an opaque DateTime value that specifies a more recent blob snapshot to be compared against a previous snapshot (previous_snapshot).

- **start_range** (*int*) – Start of byte range to use for getting different page ranges. If no `end_range` is given, all bytes after the `start_range` will be searched. Pages must be aligned with 512-byte boundaries, the start offset must be a modulus of 512 and the end offset must be a modulus of 512-1. Examples of valid byte ranges are 0-511, 512-, etc.
- **end_range** (*int*) – End of byte range to use for getting different page ranges. If `end_range` is given, `start_range` must be provided. This range will return valid page ranges for from the offset start up to offset end. Pages must be aligned with 512-byte boundaries, the start offset must be a modulus of 512 and the end offset must be a modulus of 512-1. Examples of valid byte ranges are 0-511, 512-, etc.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A `DateTime` value. Azure expects the date value passed in to be UTC. If `timezone` is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without `timezone` info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A `DateTime` value. Azure expects the date value passed in to be UTC. If `timezone` is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without `timezone` info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource's ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns A list of different Page Ranges for the Page Blob.

Return type list of `PageRange`

incremental_copy_blob (*container_name, blob_name, copy_source, metadata=None, destination_if_modified_since=None, destination_if_unmodified_since=None, destination_if_match=None, destination_if_none_match=None, destination_lease_id=None, source_lease_id=None, timeout=None*)

Copies an incremental copy of a blob asynchronously. This operation returns a copy operation properties object, including a copy ID you can use to check or abort the copy operation. The Blob service copies blobs on a best-effort basis.

The source blob for an incremental copy operation must be a page blob. Call `get_blob_properties` on the destination blob to check the status of the copy operation. The final blob will be committed when the copy completes.

Parameters

- **container_name** (*str*) – Name of the destination container. The container must exist.
- **blob_name** (*str*) – Name of the destination blob. If the destination blob exists, it will be overwritten. Otherwise, it will be created.
- **copy_source** (*str*) – A URL of up to 2 KB in length that specifies an Azure page blob. The value should be URL-encoded as it would appear in a

request URI. The copy source must be a snapshot and include a valid SAS token or be public. Example: <https://myaccount.blob.core.windows.net/mycontainer/myblob?snapshot=<DateTime>&sastoken>

- **metadata** (*A dict mapping str to str.*) – Name-value pairs associated with the blob as metadata. If no name-value pairs are specified, the operation will copy the metadata from the source blob or file to the destination blob. If one or more name-value pairs are specified, the destination blob is created with the specified metadata, and metadata is not copied from the source blob or file.
- **destination_if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this conditional header to copy the blob only if the destination blob has been modified since the specified date/time. If the destination blob has not been modified, the Blob service returns status code 412 (Precondition Failed).
- **destination_if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this conditional header to copy the blob only if the destination blob has not been modified since the specified ate/time. If the destination blob has been modified, the Blob service returns status code 412 (Precondition Failed).
- **destination_if_match** (*ETag*) – An ETag value, or the wildcard character (*). Specify an ETag value for this conditional header to copy the blob only if the specified ETag value matches the ETag value for an existing destination blob. If the ETag for the destination blob does not match the ETag specified for If-Match, the Blob service returns status code 412 (Precondition Failed).
- **destination_if_none_match** (*ETag*) – An ETag value, or the wildcard character (*). Specify an ETag value for this conditional header to copy the blob only if the specified ETag value does not match the ETag value for the destination blob. Specify the wildcard character (*) to perform the operation only if the destination blob does not exist. If the specified condition isn't met, the Blob service returns status code 412 (Precondition Failed).
- **destination_lease_id** (*str*) – The lease ID specified for this header must match the lease ID of the destination blob. If the request does not include the lease ID or it is not valid, the operation fails with status code 412 (Precondition Failed).
- **source_lease_id** (*str*) – Specify this to perform the Copy Blob operation only if the lease ID given matches the active lease ID of the source blob.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns Copy operation properties such as status, source, and ID.

Return type *CopyProperties*

resize_blob (*container_name, blob_name, content_length, lease_id=None, if_modified_since=None, if_unmodified_since=None, if_match=None, if_none_match=None, timeout=None*)

Resizes a page blob to the specified size. If the specified value is less than the current size of the blob, then all pages above the specified value are cleared.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **content_length** (*int*) – Size to resize blob to.

- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns ETag and last modified properties for the updated Page Blob

Return type *ResourceProperties*

set_premium_page_blob_tier (*container_name, blob_name, premium_page_blob_tier, timeout=None*)

Sets the page blob tiers on the blob. This API is only supported for page blobs on premium accounts.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of blob to update.
- **premium_page_blob_tier** (*PremiumPageBlobTier*) – A page blob tier value to set the blob to. The tier correlates to the size of the blob and number of allowed IOPS. This is only applicable to page blobs on premium storage accounts.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

set_sequence_number (*container_name, blob_name, sequence_number_action, sequence_number=None, lease_id=None, if_modified_since=None, if_unmodified_since=None, if_match=None, if_none_match=None, timeout=None*)

Sets the blob sequence number.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **sequence_number_action** (*str*) – This property indicates how the service should modify the blob’s sequence number. See *SequenceNumberAction* for more information.
- **sequence_number** (*str*) – This property sets the blob’s sequence number. The sequence number is a user-controlled property that you can use to track requests and manage concurrency issues.

- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag matches the value specified.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify this header to perform the operation only if the resource’s ETag does not match the value specified. Specify the wildcard character (*) to perform the operation only if the resource does not exist, and fail the operation if it does exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns ETag and last modified properties for the updated Page Blob

Return type *ResourceProperties*

update_page (*container_name, blob_name, page, start_range, end_range, validate_content=False, lease_id=None, if_sequence_number_lte=None, if_sequence_number_lt=None, if_sequence_number_eq=None, if_modified_since=None, if_unmodified_since=None, if_match=None, if_none_match=None, timeout=None*)

Updates a range of pages.

Parameters

- **container_name** (*str*) – Name of existing container.
- **blob_name** (*str*) – Name of existing blob.
- **page** (*bytes*) – Content of the page.
- **start_range** (*int*) – Start of byte range to use for writing to a section of the blob. Pages must be aligned with 512-byte boundaries, the start offset must be a modulus of 512 and the end offset must be a modulus of 512-1. Examples of valid byte ranges are 0-511, 512-1023, etc.
- **end_range** (*int*) – End of byte range to use for writing to a section of the blob. Pages must be aligned with 512-byte boundaries, the start offset must be a modulus of 512 and the end offset must be a modulus of 512-1. Examples of valid byte ranges are 0-511, 512-1023, etc.
- **validate_content** (*bool*) – If true, calculates an MD5 hash of the page content. The storage service checks the hash of the content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the blob.
- **lease_id** (*str*) – Required if the blob has an active lease.
- **if_sequence_number_lte** (*int*) – If the blob’s sequence number is less than or equal to the specified value, the request proceeds; otherwise it fails.

- **if_sequence_number_lt** (*int*) – If the blob’s sequence number is less than the specified value, the request proceeds; otherwise it fails.
- **if_sequence_number_eq** (*int*) – If the blob’s sequence number is equal to the specified value, the request proceeds; otherwise it fails.
- **if_modified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has been modified since the specified time.
- **if_unmodified_since** (*datetime*) – A DateTime value. Azure expects the date value passed in to be UTC. If timezone is included, any non-UTC datetimes will be converted to UTC. If a date is passed in without timezone info, it is assumed to be UTC. Specify this header to perform the operation only if the resource has not been modified since the specified date/time.
- **if_match** (*str*) – An ETag value, or the wildcard character (*). Specify an ETag value for this conditional header to write the page only if the blob’s ETag value matches the value specified. If the values do not match, the Blob service fails.
- **if_none_match** (*str*) – An ETag value, or the wildcard character (*). Specify an ETag value for this conditional header to write the page only if the blob’s ETag value does not match the value specified. If the values are identical, the Blob service fails.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns ETag and last modified properties for the updated Page Blob

Return type *ResourceProperties*

Module contents

azure.storage.file package

Submodules

azure.storage.file.fileservice module

```
class azure.storage.file.fileservice.FileService (account_name=None,          ac-
count_key=None,          sas_token=None,
protocol='https',          end-
point_suffix='core.windows.net',
request_session=None,
connection_string=None,
socket_timeout=None)
```

Bases: *azure.storage.storageclient.StorageClient*

The Server Message Block (SMB) protocol is the preferred file share protocol used on premise today. The Microsoft Azure File service enables customers to leverage the availability and scalability of Azure’s Cloud Infrastructure as a Service (IaaS) SMB without having to rewrite SMB client applications.

The Azure File service also offers a compelling alternative to traditional Direct Attached Storage (DAS) and Storage Area Network (SAN) solutions, which are often complex and expensive to install, configure, and operate.

Variables

- **MAX_SINGLE_GET_SIZE** (*int*) – The size of the first range get performed by `get_file_to_*` methods if `max_connections` is greater than 1. Less data will be returned if the file is smaller than this.
- **MAX_CHUNK_GET_SIZE** (*int*) – The size of subsequent range gets performed by `get_file_to_*` methods if `max_connections` is greater than 1 and the file is larger than `MAX_SINGLE_GET_SIZE`. Less data will be returned if the remainder of the file is smaller than this. If this is set to larger than 4MB, `content_validation` will throw an error if enabled. However, if `content_validation` is not desired a size greater than 4MB may be optimal. Setting this below 4MB is not recommended.
- **MAX_RANGE_SIZE** (*int*) – The size of the ranges put by `create_file_from_*` methods. Smaller ranges may be put if there is less data provided. The maximum range size the service supports is 4MB.

Parameters

- **account_name** (*str*) – The storage account name. This is used to authenticate requests signed with an account key and to construct the storage endpoint. It is required unless a connection string is given.
- **account_key** (*str*) – The storage account key. This is used for shared key authentication.
- **sas_token** (*str*) – A shared access signature token to use to authenticate requests instead of the account key. If account key and sas token are both specified, account key will be used to sign.
- **protocol** (*str*) – The protocol to use for requests. Defaults to `https`.
- **endpoint_suffix** (*str*) – The host base component of the url, minus the account name. Defaults to `Azure (core.windows.net)`. Override this to use the China cloud (`core.chinacloudapi.cn`).
- **request_session** (*requests.Session*) – The session object to use for http requests.
- **connection_string** (*str*) – If specified, this will override all other parameters besides request session. See <http://azure.microsoft.com/en-us/documentation/articles/storage-configure-connection-string/> for the connection string format.
- **socket_timeout** (*int*) – If specified, this will override the default socket timeout. The timeout specified is in seconds. See `DEFAULT_SOCKET_TIMEOUT` in `_constants.py` for the default value.

MAX_CHUNK_GET_SIZE = 8388608

MAX_RANGE_SIZE = 4194304

MAX_SINGLE_GET_SIZE = 33554432

abort_copy_file (*share_name, directory_name, file_name, copy_id, timeout=None*)

Aborts a pending `copy_file` operation, and leaves a destination file with zero length and full metadata.

Parameters

- **share_name** (*str*) – Name of destination share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of destination file.

- **copy_id** (*str*) – Copy identifier provided in the copy.id of the original copy_file operation.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

clear_range (*share_name, directory_name, file_name, start_range, end_range, timeout=None*)

Clears the specified range and releases the space used in storage for that range.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of existing file.
- **start_range** (*int*) – Start of byte range to use for clearing a section of the file. The range can be up to 4 MB in size. The start_range and end_range params are inclusive. Ex: start_range=0, end_range=511 will download first 512 bytes of file.
- **end_range** (*int*) – End of byte range to use for clearing a section of the file. The range can be up to 4 MB in size. The start_range and end_range params are inclusive. Ex: start_range=0, end_range=511 will download first 512 bytes of file.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

copy_file (*share_name, directory_name, file_name, copy_source, metadata=None, timeout=None*)

Copies a file asynchronously. This operation returns a copy operation properties object, including a copy ID you can use to check or abort the copy operation. The File service copies files on a best-effort basis.

If the destination file exists, it will be overwritten. The destination file cannot be modified while the copy operation is in progress.

Parameters

- **share_name** (*str*) – Name of the destination share. The share must exist.
- **directory_name** (*str*) – Name of the destination directory. The directory must exist.
- **file_name** (*str*) – Name of the destination file. If the destination file exists, it will be overwritten. Otherwise, it will be created.
- **copy_source** (*str*) – A URL of up to 2 KB in length that specifies an Azure file or blob. The value should be URL-encoded as it would appear in a request URI. If the source is in another account, the source must either be public or must be authenticated via a shared access signature. If the source is public, no authentication is required. Examples: <https://myaccount.file.core.windows.net/myshare/mydir/myfile> <https://otheraccount.file.core.windows.net/myshare/mydir/myfile?sastoken>
- **metadata** (*A dict mapping str to str.*) – Name-value pairs associated with the file as metadata. If no name-value pairs are specified, the operation will copy the metadata from the source blob or file to the destination file. If one or more name-value pairs are specified, the destination file is created with the specified metadata, and the metadata is not copied from the source blob or file.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns Copy operation properties such as status, source, and ID.

Return type *CopyProperties*

create_directory (*share_name, directory_name, metadata=None, fail_on_exist=False, timeout=None*)

Creates a new directory under the specified share or parent directory. If the directory with the same name

already exists, the operation fails on the service. By default, the exception is swallowed by the client. To expose the exception, specify True for `fail_on_exists`.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – Name of directory to create, including the path to the parent directory.
- **metadata** (*dict of str to str:*) – A dict with name_value pairs to associate with the share as metadata. Example: {‘Category’:‘test’}
- **fail_on_exist** (*bool*) – specify whether to throw an exception when the directory exists. False by default.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns True if directory is created, False if directory already exists.

Return type bool

create_file (*share_name, directory_name, file_name, content_length, content_settings=None, metadata=None, timeout=None*)

Creates a new file.

See `create_file_from_*` for high level functions that handle the creation and upload of large files with automatic chunking and progress notifications.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of file to create or update.
- **content_length** (*int*) – Length of the file in bytes.
- **content_settings** (*ContentSettings*) – ContentSettings object used to set file properties.
- **metadata** (*a dict mapping str to str*) – Name-value pairs associated with the file as metadata.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

create_file_from_bytes (*share_name, directory_name, file_name, file, index=0, count=None, content_settings=None, metadata=None, validate_content=False, progress_callback=None, max_connections=2, timeout=None*)

Creates a new file from an array of bytes, or updates the content of an existing file, with automatic chunking and progress notifications.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of file to create or update.
- **file** (*str*) – Content of file as an array of bytes.
- **index** (*int*) – Start index in the array of bytes.
- **count** (*int*) – Number of bytes to upload. Set to None or negative value to upload all bytes starting from index.

- **content_settings** (*ContentSettings*) – ContentSettings object used to set file properties.
- **metadata** (*a dict mapping str to str*) – Name-value pairs associated with the file as metadata.
- **validate_content** (*bool*) – If true, calculates an MD5 hash for each range of the file. The storage service checks the hash of the content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the file.
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature function(current, total) where current is the number of bytes transferred so far and total is the size of the file, or None if the total size is unknown.
- **max_connections** (*int*) – Maximum number of parallel connections to use.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

create_file_from_path(*share_name, directory_name, file_name, local_file_path, content_settings=None, metadata=None, validate_content=False, progress_callback=None, max_connections=2, timeout=None*)

Creates a new azure file from a local file path, or updates the content of an existing file, with automatic chunking and progress notifications.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of file to create or update.
- **local_file_path** (*str*) – Path of the local file to upload as the file content.
- **content_settings** (*ContentSettings*) – ContentSettings object used for setting file properties.
- **metadata** (*a dict mapping str to str*) – Name-value pairs associated with the file as metadata.
- **validate_content** (*bool*) – If true, calculates an MD5 hash for each range of the file. The storage service checks the hash of the content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the file.
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature function(current, total) where current is the number of bytes transferred so far and total is the size of the file, or None if the total size is unknown.
- **max_connections** (*int*) – Maximum number of parallel connections to use.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

create_file_from_stream(*share_name, directory_name, file_name, stream, count, content_settings=None, metadata=None, validate_content=False, progress_callback=None, max_connections=2, timeout=None*)

Creates a new file from a file/stream, or updates the content of an existing file, with automatic chunking and progress notifications.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of file to create or update.
- **stream** (*io.IOBase*) – Opened file/stream to upload as the file content.
- **count** (*int*) – Number of bytes to read from the stream. This is required, a file cannot be created if the count is unknown.
- **content_settings** (*ContentSettings*) – ContentSettings object used to set file properties.
- **metadata** (*a dict mapping str to str*) – Name-value pairs associated with the file as metadata.
- **validate_content** (*bool*) – If true, calculates an MD5 hash for each range of the file. The storage service checks the hash of the content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the file.
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature function(current, total) where current is the number of bytes transferred so far and total is the size of the file, or None if the total size is unknown.
- **max_connections** (*int*) – Maximum number of parallel connections to use. Note that parallel upload requires the stream to be seekable.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

create_file_from_text (*share_name, directory_name, file_name, text, encoding='utf-8', content_settings=None, metadata=None, validate_content=False, timeout=None*)

Creates a new file from str/unicode, or updates the content of an existing file, with automatic chunking and progress notifications.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of file to create or update.
- **text** (*str*) – Text to upload to the file.
- **encoding** (*str*) – Python encoding to use to convert the text to bytes.
- **content_settings** (*ContentSettings*) – ContentSettings object used to set file properties.
- **metadata** (*a dict mapping str to str*) – Name-value pairs associated with the file as metadata.
- **validate_content** (*bool*) – If true, calculates an MD5 hash for each range of the file. The storage service checks the hash of the content that has arrived with the hash that

was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the file.

- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

create_share (*share_name, metadata=None, quota=None, fail_on_exist=False, timeout=None*)

Creates a new share under the specified account. If the share with the same name already exists, the operation fails on the service. By default, the exception is swallowed by the client. To expose the exception, specify True for fail_on_exists.

Parameters

- **share_name** (*str*) – Name of share to create.
- **metadata** (*a dict of str to str:*) – A dict with name_value pairs to associate with the share as metadata. Example: {'Category': 'test' }
- **quota** (*int*) – Specifies the maximum size of the share, in gigabytes. Must be greater than 0, and less than or equal to 5TB (5120).
- **fail_on_exist** (*bool*) – Specify whether to throw an exception when the share exists. False by default.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns True if share is created, False if share already exists.

Return type bool

delete_directory (*share_name, directory_name, fail_not_exist=False, timeout=None*)

Deletes the specified empty directory. Note that the directory must be empty before it can be deleted. Attempting to delete directories that are not empty will fail.

If the directory does not exist, the operation fails on the service. By default, the exception is swallowed by the client. To expose the exception, specify True for fail_not_exist.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – Name of directory to delete, including the path to the parent directory.
- **fail_not_exist** (*bool*) – Specify whether to throw an exception when the directory doesn't exist.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns True if directory is deleted, False otherwise.

Return type bool

delete_file (*share_name, directory_name, file_name, timeout=None*)

Marks the specified file for deletion. The file is later deleted during garbage collection.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of existing file.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

delete_share (*share_name*, *fail_not_exist=False*, *timeout=None*)

Marks the specified share for deletion. If the share does not exist, the operation fails on the service. By default, the exception is swallowed by the client. To expose the exception, specify True for *fail_not_exist*.

Parameters

- **share_name** (*str*) – Name of share to delete.
- **fail_not_exist** (*bool*) – Specify whether to throw an exception when the share doesn't exist. False by default.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns True if share is deleted, False share doesn't exist.

Return type bool

exists (*share_name*, *directory_name=None*, *file_name=None*, *timeout=None*)

Returns a boolean indicating whether the share exists if only share name is given. If *directory_name* is specified a boolean will be returned indicating if the directory exists. If *file_name* is specified as well, a boolean will be returned indicating if the file exists.

Parameters

- **share_name** (*str*) – Name of a share.
- **directory_name** (*str*) – The path to a directory.
- **file_name** (*str*) – Name of a file.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns A boolean indicating whether the resource exists.

Return type bool

generate_account_shared_access_signature (*resource_types*, *permission*, *expiry*, *start=None*, *ip=None*, *protocol=None*)

Generates a shared access signature for the file service. Use the returned signature with the *sas_token* parameter of the FileService.

Parameters

- **resource_types** (*ResourceTypes*) – Specifies the resource types that are accessible with the account SAS.
- **permission** (*AccountPermissions*) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy.
- **expiry** (*datetime.datetime or str*) – The time at which the shared access signature becomes invalid. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **start** (*datetime.datetime or str*) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **ip** (*str*) – Specifies an IP address or a range of IP addresses from which to accept requests. If the IP address from which the request originates does not match the IP address or address range specified on the SAS token, the request is not authenticated. For example,

specifying `sip=168.1.5.65` or `sip=168.1.5.60-168.1.5.70` on the SAS restricts the request to those IP addresses.

- **protocol** (*str*) – Specifies the protocol permitted for a request made. Possible values are both HTTPS and HTTP (`https,http`) or HTTPS only (`https`). The default value is `https,http`. Note that HTTP only is not a permitted value.

Returns A Shared Access Signature (sas) token.

Return type *str*

generate_file_shared_access_signature (*share_name*, *directory_name=None*, *file_name=None*, *permission=None*, *expiry=None*, *start=None*, *id=None*, *ip=None*, *protocol=None*, *cache_control=None*, *content_disposition=None*, *content_encoding=None*, *content_language=None*, *content_type=None*)

Generates a shared access signature for the file. Use the returned signature with the `sas_token` parameter of `FileService`.

Parameters

- **share_name** (*str*) – Name of share.
- **directory_name** (*str*) – Name of directory. SAS tokens cannot be created for directories, so this parameter should only be present if `file_name` is provided.
- **file_name** (*str*) – Name of file.
- **permission** (`FilePermissions`) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Permissions must be ordered read, create, write, delete, list. Required unless an `id` is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy.
- **expiry** (*datetime.datetime* or *str*) – The time at which the shared access signature becomes invalid. Required unless an `id` is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **start** (*datetime.datetime* or *str*) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **id** (*str*) – A unique value up to 64 characters in length that correlates to a stored access policy. To create a stored access policy, use `set_file_service_properties`.
- **ip** (*str*) – Specifies an IP address or a range of IP addresses from which to accept requests. If the IP address from which the request originates does not match the IP address or address range specified on the SAS token, the request is not authenticated. For example, specifying `sip=168.1.5.65` or `sip=168.1.5.60-168.1.5.70` on the SAS restricts the request to those IP addresses.
- **protocol** (*str*) – Specifies the protocol permitted for a request made. Possible values are both HTTPS and HTTP (`https,http`) or HTTPS only (`https`). The default value is `https,http`. Note that HTTP only is not a permitted value.
- **cache_control** (*str*) – Response header value for Cache-Control when resource is accessed using this shared access signature.

- **content_disposition** (*str*) – Response header value for Content-Disposition when resource is accessed using this shared access signature.
- **content_encoding** (*str*) – Response header value for Content-Encoding when resource is accessed using this shared access signature.
- **content_language** (*str*) – Response header value for Content-Language when resource is accessed using this shared access signature.
- **content_type** (*str*) – Response header value for Content-Type when resource is accessed using this shared access signature.

Returns A Shared Access Signature (sas) token.

Return type `str`

generate_share_shared_access_signature (*share_name*, *permission=None*, *expiry=None*, *start=None*, *id=None*, *ip=None*, *protocol=None*, *cache_control=None*, *content_disposition=None*, *content_encoding=None*, *content_language=None*, *content_type=None*)

Generates a shared access signature for the share. Use the returned signature with the `sas_token` parameter of `FileService`.

Parameters

- **share_name** (*str*) – Name of share.
- **permission** (`SharePermissions`) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Permissions must be ordered read, create, write, delete, list. Required unless an `id` is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy.
- **expiry** (`datetime.datetime` or *str*) – The time at which the shared access signature becomes invalid. Required unless an `id` is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **start** (`datetime.datetime` or *str*) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **id** (*str*) – A unique value up to 64 characters in length that correlates to a stored access policy. To create a stored access policy, use `set_share_acl()`.
- **ip** (*str*) – Specifies an IP address or a range of IP addresses from which to accept requests. If the IP address from which the request originates does not match the IP address or address range specified on the SAS token, the request is not authenticated. For example, specifying `sip=168.1.5.65` or `sip=168.1.5.60-168.1.5.70` on the SAS restricts the request to those IP addresses.
- **protocol** (*str*) – Specifies the protocol permitted for a request made. Possible values are both HTTPS and HTTP (`https,http`) or HTTPS only (`https`). The default value is `https,http`. Note that HTTP only is not a permitted value.
- **cache_control** (*str*) – Response header value for Cache-Control when resource is accessed using this shared access signature.

- **content_disposition** (*str*) – Response header value for Content-Disposition when resource is accessed using this shared access signature.
- **content_encoding** (*str*) – Response header value for Content-Encoding when resource is accessed using this shared access signature.
- **content_language** (*str*) – Response header value for Content-Language when resource is accessed using this shared access signature.
- **content_type** (*str*) – Response header value for Content-Type when resource is accessed using this shared access signature.

Returns A Shared Access Signature (sas) token.

Return type *str*

get_directory_metadata (*share_name, directory_name, timeout=None*)

Returns all user-defined metadata for the specified directory.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns A dictionary representing the directory metadata name, value pairs.

Return type a dict mapping *str* to *str*

get_directory_properties (*share_name, directory_name, timeout=None*)

Returns all user-defined metadata and system properties for the specified directory. The data returned does not include the directory's list of files.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to an existing directory.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns properties for the specified directory within a directory object.

Return type *Directory*

get_file_metadata (*share_name, directory_name, file_name, timeout=None*)

Returns all user-defined metadata for the specified file.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of existing file.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns A dictionary representing the file metadata name, value pairs.

Return type dict mapping *str* to *str*.

get_file_properties (*share_name, directory_name, file_name, timeout=None*)

Returns all user-defined metadata, standard HTTP properties, and system properties for the file. Returns an instance of *File* with *FileProperties* and a metadata dict.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of existing file.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns a file object including properties and metadata.

Return type *File*

get_file_service_properties (*timeout=None*)

Gets the properties of a storage account’s File service, including Azure Storage Analytics.

Parameters **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns The file service properties.

Return type *ServiceProperties*

get_file_to_bytes (*share_name, directory_name, file_name, start_range=None, end_range=None, validate_content=False, progress_callback=None, max_connections=2, timeout=None*)

Downloads a file as an array of bytes, with automatic chunking and progress notifications. Returns an instance of *File* with properties, metadata, and content.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of existing file.
- **start_range** (*int*) – Start of byte range to use for downloading a section of the file. If no *end_range* is given, all bytes after the *start_range* will be downloaded. The *start_range* and *end_range* params are inclusive. Ex: *start_range=0, end_range=511* will download first 512 bytes of file.
- **end_range** (*int*) – End of byte range to use for downloading a section of the file. If *end_range* is given, *start_range* must be provided. The *start_range* and *end_range* params are inclusive. Ex: *start_range=0, end_range=511* will download first 512 bytes of file.
- **validate_content** (*bool*) – If set to true, validates an MD5 hash for each retrieved portion of the file. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that the service will only return transactional MD5s for chunks 4MB or less so the first get request will be of size *self.MAX_CHUNK_GET_SIZE* instead of *self.MAX_SINGLE_GET_SIZE*. If *self.MAX_CHUNK_GET_SIZE* was set to greater than 4MB an error will be thrown. As computing the MD5 takes processing time and more requests will need to be done due to the reduced chunk size there may be some increase in latency.
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature *function(current, total)* where *current* is the number of bytes transferred so far, and *total* is the size of the file if known.
- **max_connections** (*int*) – If set to 2 or greater, an initial get will be done for the first *self.MAX_SINGLE_GET_SIZE* bytes of the file. If this is the entire file, the method returns at this point. If it is not, it will download the remaining data parallel using the number of threads equal to *max_connections*. Each chunk will be of size

`self.MAX_CHUNK_GET_SIZE`. If set to 1, a single large get request will be done. This is not generally recommended but available if very few threads should be used, network requests are very expensive, or a non-seekable stream prevents parallel download. This may also be valuable if the file is being concurrently modified to enforce atomicity or if many files are expected to be empty as an extra request is required for empty files if `max_connections` is greater than 1.

- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

Returns A File with properties, content, and metadata.

Return type *File*

```
get_file_to_path(share_name, directory_name, file_name, file_path, open_mode='wb',
                  start_range=None, end_range=None, validate_content=False,
                  progress_callback=None, max_connections=2, timeout=None)
```

Downloads a file to a file path, with automatic chunking and progress notifications. Returns an instance of File with properties and metadata.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of existing file.
- **file_path** (*str*) – Path of file to write to.
- **open_mode** (*str*) – Mode to use when opening the file. Note that specifying append only `open_mode` prevents parallel download. So, `max_connections` must be set to 1 if this `open_mode` is used.
- **start_range** (*int*) – Start of byte range to use for downloading a section of the file. If no `end_range` is given, all bytes after the `start_range` will be downloaded. The `start_range` and `end_range` params are inclusive. Ex: `start_range=0, end_range=511` will download first 512 bytes of file.
- **end_range** (*int*) – End of byte range to use for downloading a section of the file. If `end_range` is given, `start_range` must be provided. The `start_range` and `end_range` params are inclusive. Ex: `start_range=0, end_range=511` will download first 512 bytes of file.
- **validate_content** (*bool*) – If set to true, validates an MD5 hash for each retrieved portion of the file. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that the service will only return transactional MD5s for chunks 4MB or less so the first get request will be of size `self.MAX_CHUNK_GET_SIZE` instead of `self.MAX_SINGLE_GET_SIZE`. If `self.MAX_CHUNK_GET_SIZE` was set to greater than 4MB an error will be thrown. As computing the MD5 takes processing time and more requests will need to be done due to the reduced chunk size there may be some increase in latency.
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature `function(current, total)` where `current` is the number of bytes transferred so far, and `total` is the size of the file if known.
- **max_connections** (*int*) – If set to 2 or greater, an initial get will be done for the first `self.MAX_SINGLE_GET_SIZE` bytes of the file. If this is the entire file, the method returns at this point. If it is not, it will download the remaining data parallel using the number of threads equal to `max_connections`. Each chunk will be of size `self.MAX_CHUNK_GET_SIZE`. If set to 1, a single large get request will be done. This

is not generally recommended but available if very few threads should be used, network requests are very expensive, or a non-seekable stream prevents parallel download. This may also be valuable if the file is being concurrently modified to enforce atomicity or if many files are expected to be empty as an extra request is required for empty files if `max_connections` is greater than 1.

- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

Returns A File with properties and metadata.

Return type *File*

get_file_to_stream (*share_name, directory_name, file_name, stream, start_range=None, end_range=None, validate_content=False, progress_callback=None, max_connections=2, timeout=None*)

Downloads a file to a stream, with automatic chunking and progress notifications. Returns an instance of *File* with properties and metadata.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of existing file.
- **stream** (*io.IOBase*) – Opened file/stream to write to.
- **start_range** (*int*) – Start of byte range to use for downloading a section of the file. If no `end_range` is given, all bytes after the `start_range` will be downloaded. The `start_range` and `end_range` params are inclusive. Ex: `start_range=0, end_range=511` will download first 512 bytes of file.
- **end_range** (*int*) – End of byte range to use for downloading a section of the file. If `end_range` is given, `start_range` must be provided. The `start_range` and `end_range` params are inclusive. Ex: `start_range=0, end_range=511` will download first 512 bytes of file.
- **validate_content** (*bool*) – If set to true, validates an MD5 hash for each retrieved portion of the file. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that the service will only return transactional MD5s for chunks 4MB or less so the first get request will be of size `self.MAX_CHUNK_GET_SIZE` instead of `self.MAX_SINGLE_GET_SIZE`. If `self.MAX_CHUNK_GET_SIZE` was set to greater than 4MB an error will be thrown. As computing the MD5 takes processing time and more requests will need to be done due to the reduced chunk size there may be some increase in latency.
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature `function(current, total)` where `current` is the number of bytes transferred so far, and `total` is the size of the file if known.
- **max_connections** (*int*) – If set to 2 or greater, an initial get will be done for the first `self.MAX_SINGLE_GET_SIZE` bytes of the file. If this is the entire file, the method returns at this point. If it is not, it will download the remaining data parallel using the number of threads equal to `max_connections`. Each chunk will be of size `self.MAX_CHUNK_GET_SIZE`. If set to 1, a single large get request will be done. This is not generally recommended but available if very few threads should be used, network requests are very expensive, or a non-seekable stream prevents parallel download. This may also be valuable if the file is being concurrently modified to enforce atomicity or

if many files are expected to be empty as an extra request is required for empty files if `max_connections` is greater than 1.

- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

Returns A File with properties and metadata.

Return type *File*

get_file_to_text (*share_name, directory_name, file_name, encoding='utf-8', start_range=None, end_range=None, validate_content=False, progress_callback=None, max_connections=2, timeout=None*)

Downloads a file as unicode text, with automatic chunking and progress notifications. Returns an instance of `File` with properties, metadata, and content.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of existing file.
- **encoding** (*str*) – Python encoding to use when decoding the file data.
- **start_range** (*int*) – Start of byte range to use for downloading a section of the file. If no `end_range` is given, all bytes after the `start_range` will be downloaded. The `start_range` and `end_range` params are inclusive. Ex: `start_range=0, end_range=511` will download first 512 bytes of file.
- **end_range** (*int*) – End of byte range to use for downloading a section of the file. If `end_range` is given, `start_range` must be provided. The `start_range` and `end_range` params are inclusive. Ex: `start_range=0, end_range=511` will download first 512 bytes of file.
- **validate_content** (*bool*) – If set to true, validates an MD5 hash for each retrieved portion of the file. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that the service will only return transactional MD5s for chunks 4MB or less so the first get request will be of size `self.MAX_CHUNK_GET_SIZE` instead of `self.MAX_SINGLE_GET_SIZE`. If `self.MAX_CHUNK_GET_SIZE` was set to greater than 4MB an error will be thrown. As computing the MD5 takes processing time and more requests will need to be done due to the reduced chunk size there may be some increase in latency.
- **progress_callback** (*callback function in format of func(current, total)*) – Callback for progress with signature `function(current, total)` where `current` is the number of bytes transferred so far, and `total` is the size of the file if known.
- **max_connections** (*int*) – If set to 2 or greater, an initial get will be done for the first `self.MAX_SINGLE_GET_SIZE` bytes of the file. If this is the entire file, the method returns at this point. If it is not, it will download the remaining data parallel using the number of threads equal to `max_connections`. Each chunk will be of size `self.MAX_CHUNK_GET_SIZE`. If set to 1, a single large get request will be done. This is not generally recommended but available if very few threads should be used, network requests are very expensive, or a non-seekable stream prevents parallel download. This may also be valuable if the file is being concurrently modified to enforce atomicity or if many files are expected to be empty as an extra request is required for empty files if `max_connections` is greater than 1.
- **timeout** (*int*) – The timeout parameter is expressed in seconds. This method may make multiple calls to the Azure service and the timeout will apply to each call individually.

Returns A File with properties, content, and metadata.

Return type *File*

get_share_acl (*share_name*, *timeout=None*)

Gets the permissions for the specified share.

Parameters

- **share_name** (*str*) – Name of existing share.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns A dictionary of access policies associated with the share.

Return type dict of str to *AccessPolicy*

get_share_metadata (*share_name*, *timeout=None*)

Returns all user-defined metadata for the specified share.

Parameters

- **share_name** (*str*) – Name of existing share.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns A dictionary representing the share metadata name, value pairs.

Return type a dict mapping str to str

get_share_properties (*share_name*, *timeout=None*)

Returns all user-defined metadata and system properties for the specified share. The data returned does not include the share's list of files or directories.

Parameters

- **share_name** (*str*) – Name of existing share.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns A Share that exposes properties and metadata.

Return type *Share*

get_share_stats (*share_name*, *timeout=None*)

Gets the approximate size of the data stored on the share, rounded up to the nearest gigabyte.

Note that this value may not include all recently created or recently resized files.

Parameters

- **share_name** (*str*) – Name of existing share.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns the approximate size of the data stored on the share.

Return type int

list_directories_and_files (*share_name*, *directory_name=None*, *num_results=None*,
marker=None, *timeout=None*, *prefix=None*)

Returns a generator to list the directories and files under the specified share. The generator will lazily follow the continuation tokens returned by the service and stop when all directories and files have been returned or num_results is reached.

If num_results is specified and the share has more than that number of files and directories, the generator will have a populated next_marker field once it finishes. This marker can be used to create a new generator if more results are desired.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **num_results** (*int*) – Specifies the maximum number of files to return, including all directory elements. If the request does not specify num_results or specifies a value greater than 5,000, the server will return up to 5,000 items. Setting num_results to a value less than or equal to zero results in error response code 400 (Bad Request).
- **marker** (*str*) – An opaque continuation token. This value can be retrieved from the next_marker field of a previous generator object if num_results was specified and that generator has finished enumerating results. If specified, this generator will begin returning results from the point where the previous generator stopped.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.
- **prefix** (*str*) – List only the files and/or directories with the given prefix.

list_ranges (*share_name, directory_name, file_name, start_range=None, end_range=None, timeout=None*)

Retrieves the valid ranges for a file.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of existing file.
- **start_range** (*int*) – Specifies the start offset of bytes over which to list ranges. The start_range and end_range params are inclusive. Ex: start_range=0, end_range=511 will download first 512 bytes of file.
- **end_range** (*int*) – Specifies the end offset of bytes over which to list ranges. The start_range and end_range params are inclusive. Ex: start_range=0, end_range=511 will download first 512 bytes of file.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns a list of valid ranges

Return type a list of *FileRange*

list_shares (*prefix=None, marker=None, num_results=None, include_metadata=False, timeout=None*)

Returns a generator to list the shares under the specified account. The generator will lazily follow the continuation tokens returned by the service and stop when all shares have been returned or num_results is reached.

If num_results is specified and the account has more than that number of shares, the generator will have a populated next_marker field once it finishes. This marker can be used to create a new generator if more results are desired.

Parameters

- **prefix** (*str*) – Filters the results to return only shares whose names begin with the specified prefix.
- **num_results** (*int*) – Specifies the maximum number of shares to return.
- **include_metadata** (*bool*) – Specifies that share metadata be returned in the response.

- **marker** (*str*) – An opaque continuation token. This value can be retrieved from the `next_marker` field of a previous generator object if `num_results` was specified and that generator has finished enumerating results. If specified, this generator will begin returning results from the point where the previous generator stopped.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

make_file_url (*share_name, directory_name, file_name, protocol=None, sas_token=None*)
Creates the url to access a file.

Parameters

- **share_name** (*str*) – Name of share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of file.
- **protocol** (*str*) – Protocol to use: ‘http’ or ‘https’. If not specified, uses the protocol specified when FileService was initialized.
- **sas_token** (*str*) – Shared access signature token created with `generate_shared_access_signature`.

Returns file access URL.

Return type str

resize_file (*share_name, directory_name, file_name, content_length, timeout=None*)

Resizes a file to the specified size. If the specified byte value is less than the current size of the file, then all ranges above the specified byte value are cleared.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of existing file.
- **content_length** (*int*) – The length to resize the file to.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

set_directory_metadata (*share_name, directory_name, metadata=None, timeout=None*)

Sets one or more user-defined name-value pairs for the specified directory. Each call to this operation replaces all existing metadata attached to the directory. To remove all metadata from the directory, call this operation with no metadata dict.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **metadata** (*A dict mapping str to str.*) – A dict containing name-value pairs to associate with the directory as metadata. Example: {‘category’:‘test’}
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

set_file_metadata (*share_name, directory_name, file_name, metadata=None, timeout=None*)

Sets user-defined metadata for the specified file as one or more name-value pairs.

Parameters

- **share_name** (*str*) – Name of existing share.

- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of existing file.
- **metadata** (*dict mapping str to str*) – Dict containing name and value pairs. Each call to this operation replaces all existing metadata attached to the file. To remove all metadata from the file, call this operation with no metadata headers.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

set_file_properties (*share_name, directory_name, file_name, content_settings, timeout=None*)

Sets system properties on the file. If one property is set for the `content_settings`, all properties will be overridden.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of existing file.
- **content_settings** (`ContentSettings`) – `ContentSettings` object used to set the file properties.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

set_file_service_properties (*hour_metrics=None, minute_metrics=None, cors=None, timeout=None*)

Sets the properties of a storage account's File service, including Azure Storage Analytics. If an element (ex `HourMetrics`) is left as `None`, the existing settings on the service for that functionality are preserved.

Parameters

- **hour_metrics** (`Metrics`) – The hour metrics settings provide a summary of request statistics grouped by API in hourly aggregates for files.
- **minute_metrics** (`Metrics`) – The minute metrics settings provide request statistics for each minute for files.
- **cors** (list of `CorsRule`) – You can include up to five `CorsRule` elements in the list. If an empty list is specified, all CORS rules will be deleted, and CORS will be disabled for the service.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

set_share_acl (*share_name, signed_identifiers=None, timeout=None*)

Sets the permissions for the specified share or stored access policies that may be used with Shared Access Signatures.

Parameters

- **share_name** (*str*) – Name of existing share.
- **signed_identifiers** (dict of *str* to `AccessPolicy`) – A dictionary of access policies to associate with the share. The dictionary may contain up to 5 elements. An empty dictionary will clear the access policies set on the service.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

set_share_metadata (*share_name, metadata=None, timeout=None*)

Sets one or more user-defined name-value pairs for the specified share. Each call to this operation replaces all existing metadata attached to the share. To remove all metadata from the share, call this operation with no metadata dict.

Parameters

- **share_name** (*str*) – Name of existing share.
- **metadata** (*a dict mapping str to str*) – A dict containing name-value pairs to associate with the share as metadata. Example: { 'category': 'test' }
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

set_share_properties (*share_name, quota, timeout=None*)

Sets service-defined properties for the specified share.

Parameters

- **share_name** (*str*) – Name of existing share.
- **quota** (*int*) – Specifies the maximum size of the share, in gigabytes. Must be greater than 0, and less than or equal to 5 TB (5120 GB).
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

update_range (*share_name, directory_name, file_name, data, start_range, end_range, validate_content=False, timeout=None*)

Writes the bytes specified by the request body into the specified range.

Parameters

- **share_name** (*str*) – Name of existing share.
- **directory_name** (*str*) – The path to the directory.
- **file_name** (*str*) – Name of existing file.
- **data** (*bytes*) – Content of the range.
- **start_range** (*int*) – Start of byte range to use for updating a section of the file. The range can be up to 4 MB in size. The start_range and end_range params are inclusive. Ex: start_range=0, end_range=511 will download first 512 bytes of file.
- **end_range** (*int*) – End of byte range to use for updating a section of the file. The range can be up to 4 MB in size. The start_range and end_range params are inclusive. Ex: start_range=0, end_range=511 will download first 512 bytes of file.
- **validate_content** (*bool*) – If true, calculates an MD5 hash of the page content. The storage service checks the hash of the content that has arrived with the hash that was sent. This is primarily valuable for detecting bitflips on the wire if using http instead of https as https (the default) will already validate. Note that this MD5 hash is not stored with the file.
- **timeout** (*int*) – The timeout parameter is expressed in seconds.

azure.storage.file.models module

```
class azure.storage.file.models.ContentSettings (content_type=None,
                                                content_encoding=None,
                                                content_language=None,
                                                content_disposition=None,
                                                cache_control=None,
                                                content_md5=None)
```

Bases: object

Used to store the content settings of a file.

Variables

- **content_type** (*str*) – The content type specified for the file. If no content type was specified, the default content type is application/octet-stream.
- **content_encoding** (*str*) – If content_encoding has previously been set for the file, that value is stored.
- **content_language** (*str*) – If content_language has previously been set for the file, that value is stored.
- **content_disposition** (*str*) – content_disposition conveys additional information about how to process the response payload, and also can be used to attach additional meta-data. If content_disposition has previously been set for the file, that value is stored.
- **cache_control** (*str*) – If cache_control has previously been set for the file, that value is stored.
- **content_md5** (*str*) – If the content_md5 has been set for the file, this response header is stored so that the client can check for message content integrity.

class azure.storage.file.models.**CopyProperties**

Bases: object

File Copy Properties.

Variables

- **id** (*str*) – String identifier for the last attempted Copy File operation where this file was the destination file. This header does not appear if this file has never been the destination in a Copy File operation, or if this file has been modified after a concluded Copy File operation using Set File Properties or Put File.
- **source** (*str*) – URL up to 2 KB in length that specifies the source file used in the last attempted Copy File operation where this file was the destination file. This header does not appear if this file has never been the destination in a Copy File operation, or if this file has been modified after a concluded Copy File operation using Set File Properties or Put File.
- **status** (*str*) –

State of the copy operation identified by Copy ID, with these values:

success: Copy completed successfully.

pending: Copy is in progress. Check copy_status_description if intermittent, non-fatal errors impede copy progress but don't cause failure.

aborted: Copy was ended by Abort Copy File.

failed: Copy failed. See copy_status_description for failure details.

- **progress** (*str*) – Contains the number of bytes copied and the total bytes in the source in the last attempted Copy File operation where this file was the destination file. Can show between 0 and Content-Length bytes copied.
- **completion_time** (*datetime*) – Conclusion time of the last attempted Copy File operation where this file was the destination file. This value can specify the time of a completed, aborted, or failed copy attempt.
- **status_description** (*str*) – Only appears when x-ms-copy-status is failed or pending. Describes cause of fatal or non-fatal copy operation failure.

class azure.storage.file.models.**Directory** (*name=None, props=None, metadata=None*)

Bases: object

Directory class.

Variables

- **name** (*str*) – The name of the directory.
- **properties** (*DirectoryProperties*) – System properties for the directory.
- **metadata** (*dict mapping str to str*) – A dict containing name-value pairs associated with the directory as metadata. This var is set to None unless the include=metadata param was included for the list directory operation. If this parameter was specified but the directory has no metadata, metadata will be set to an empty dictionary.

class `azure.storage.file.models.DirectoryProperties`

Bases: `object`

File directory's properties class.

Variables

- **last_modified** (*datetime*) – A datetime object representing the last time the directory was modified.
- **etag** (*str*) – The ETag contains a value that you can use to perform operations conditionally.
- **server_encrypted** (*bool*) – Set to true if the directory metadata is encrypted on the server.

class `azure.storage.file.models.File` (*name=None, content=None, props=None, metadata=None*)

Bases: `object`

File class.

Variables

- **name** (*str*) – The name of the file.
- **content** (*str or bytes*) – File content.
- **properties** (*FileProperties*) – System properties for the file.
- **metadata** (*dict mapping str to str*) – A dict containing name-value pairs associated with the file as metadata. This var is set to None unless the include=metadata param was included for the list file operation. If this parameter was specified but the file has no metadata, metadata will be set to an empty dictionary.

class `azure.storage.file.models.FilePermissions` (*read=False, create=False, write=False, delete=False, _str=None*)

Bases: `object`

FilePermissions class to be used with `generate_file_shared_access_signature()` API.

Variables

- **FilePermissions.CREATE** (*FilePermissions*) – Create a new file or copy a file to a new file.
- **FilePermissions.DELETE** (*FilePermissions*) – Delete the file.
- **FilePermissions.READ** (*FilePermissions*) – Read the content, properties, metadata. Use the file as the source of a copy operation.
- **FilePermissions.WRITE** (*FilePermissions*) – Create or write content, properties, metadata. Resize the file. Use the file as the destination of a copy operation within the same account.

Parameters

- **read** (*bool*) – Read the content, properties, metadata. Use the file as the source of a copy operation.
- **create** (*bool*) – Create a new file or copy a file to a new file.
- **write** (*bool*) – Create or write content, properties, metadata. Resize the file. Use the file as the destination of a copy operation within the same account.
- **delete** (*bool*) – Delete the file.
- **_str** (*str*) – A string representing the permissions.

CREATE = <azure.storage.file.models.FilePermissions object>

DELETE = <azure.storage.file.models.FilePermissions object>

READ = <azure.storage.file.models.FilePermissions object>

WRITE = <azure.storage.file.models.FilePermissions object>

class azure.storage.file.models.**FileProperties**

Bases: object

File Properties.

Variables

- **last_modified** (*datetime*) – A datetime object representing the last time the file was modified.
- **etag** (*str*) – The ETag contains a value that you can use to perform operations conditionally.
- **content_length** (*int*) – The length of the content returned. If the entire blob was requested, the length of blob in bytes. If a subset of the blob was requested, the length of the returned subset.
- **content_range** (*str*) – Indicates the range of bytes returned in the event that the client requested a subset of the blob.
- **content_settings** (*ContentSettings*) – Stores all the content settings for the file.
- **copy** (*CopyProperties*) – Stores all the copy properties for the file.

ivar bool server_encrypted: Set to true if the file data and application metadata are completely encrypted.

class azure.storage.file.models.**FileRange** (*start=None, end=None*)

Bases: object

File Range.

Variables

- **start** (*int*) – Byte index for start of file range.
- **end** (*int*) – Byte index for end of file range.

class azure.storage.file.models.**Share** (*name=None, props=None, metadata=None*)

Bases: object

File share class.

Variables

- **name** (*str*) – The name of the share.

- **properties** (`ShareProperties`) – System properties for the share.
- **metadata** (*dict mapping str to str*) – A dict containing name-value pairs associated with the share as metadata. This var is set to None unless the `include=metadata` param was included for the list shares operation. If this parameter was specified but the share has no metadata, metadata will be set to an empty dictionary.

class `azure.storage.file.models.SharePermissions` (*read=False, write=False, delete=False, list=False, _str=None*)

Bases: `object`

SharePermissions class to be used with `azure.storage.file.FileService.generate_share_shared_access_signature` method and for the AccessPolicies used with `azure.storage.file.FileService.set_share_acl`.

Variables

- **`FilePermissions.DELETE`** (`SharePermissions`) – Delete any file in the share. Note: You cannot grant permissions to delete a share with a service SAS. Use an account SAS instead.
- **`FilePermissions.LIST`** (`SharePermissions`) – List files and directories in the share.
- **`FilePermissions.READ`** (`SharePermissions`) – Read the content, properties or metadata of any file in the share. Use any file in the share as the source of a copy operation.
- **`FilePermissions.WRITE`** (`SharePermissions`) – For any file in the share, create or write content, properties or metadata. Resize the file. Use the file as the destination of a copy operation within the same account. Note: You cannot grant permissions to read or write share properties or metadata with a service SAS. Use an account SAS instead.

Parameters

- **`read`** (*bool*) – Read the content, properties or metadata of any file in the share. Use any file in the share as the source of a copy operation.
- **`write`** (*bool*) – For any file in the share, create or write content, properties or metadata. Resize the file. Use the file as the destination of a copy operation within the same account. Note: You cannot grant permissions to read or write share properties or metadata with a service SAS. Use an account SAS instead.
- **`delete`** (*bool*) – Delete any file in the share. Note: You cannot grant permissions to delete a share with a service SAS. Use an account SAS instead.
- **`list`** (*bool*) – List files and directories in the share.
- **`_str`** (*str*) – A string representing the permissions

DELETE = <azure.storage.file.models.SharePermissions object>

LIST = <azure.storage.file.models.SharePermissions object>

READ = <azure.storage.file.models.SharePermissions object>

WRITE = <azure.storage.file.models.SharePermissions object>

class `azure.storage.file.models.ShareProperties`

Bases: `object`

File share's properties class.

Variables

- **`last_modified`** (*datetime*) – A datetime object representing the last time the share was modified.

- **etag** (*str*) – The ETag contains a value that you can use to perform operations conditionally.
- **quote** (*int*) – Returns the current share quota in GB.

Module contents

azure.storage.queue package

Submodules

azure.storage.queue.models module

class `azure.storage.queue.models.Queue`

Bases: `object`

Queue class.

Variables

- **name** (*str*) – The name of the queue.
- **metadata** (*dict mapping str to str*) – A dict containing name-value pairs associated with the queue as metadata. This var is set to None unless the `include=metadata` param was included for the list queues operation. If this parameter was specified but the queue has no metadata, metadata will be set to an empty dictionary.

class `azure.storage.queue.models.QueueMessage`

Bases: `object`

Queue message class.

Variables

- **id** (*str*) – A GUID value assigned to the message by the Queue service that identifies the message in the queue. This value may be used together with the value of `pop_receipt` to delete a message from the queue after it has been retrieved with the get messages operation.
- **insertion_time** (*date*) – A UTC date value representing the time the messages was inserted.
- **expiration_time** (*date*) – A UTC date value representing the time the message expires.
- **dequeue_count** (*int*) – Begins with a value of 1 the first time the message is dequeued. This value is incremented each time the message is subsequently dequeued.
- **content** (*obj*) – The message content. Type is determined by the `decode_function` set on the service. Default is `str`.
- **pop_receipt** (*str*) – A receipt str which can be used together with the `message_id` element to delete a message from the queue after it has been retrieved with the get messages operation. Only returned by get messages operations. Set to None for peek messages.
- **time_next_visible** (*date*) – A UTC date value representing the time the message will next be visible. Only returned by get messages operations. Set to None for peek messages.

class `azure.storage.queue.models.QueueMessageFormat`

Encoding and decoding methods which can be used to modify how the queue service encodes and decodes queue messages. Set these to `queueservice.encode_function` and `queueservice.decode_function` to modify the behavior. The defaults are `text_xmlencode` and `text_xmldecode`, respectively.

static `binary_base64decode` (*data*)

Base64 decode to byte string.

Parameters `data` (*str*) – Data to decode to a byte string.

Returns Base64 decoded data.

Return type `str`

static `binary_base64encode` (*data*)

Base64 encode byte strings.

Parameters `data` (*str*) – Binary string to encode.

Returns Base64 encoded data.

Return type `str`

static `nodecode` (*data*)

Do no decoding.

Parameters `data` (*str*) – Data.

Returns The data passed in is returned unmodified.

Return type `str`

static `noencode` (*data*)

Do no encoding.

Parameters `data` (*str*) – Data.

Returns The data passed in is returned unmodified.

Return type `str`

static `text_base64decode` (*data*)

Base64 decode to unicode text.

Parameters `data` (*str*) – String data to decode to unicode.

Returns Base64 decoded string.

Return type `str`

static `text_base64encode` (*data*)

Base64 encode unicode text.

Parameters `data` (*str*) – String to encode.

Returns Base64 encoded string.

Return type `str`

static `text_xmldecode` (*data*)

XML decode to unicode text.

Parameters `data` (*str*) – Data to decode to unicode.

Returns XML decoded data.

Return type `str`

static text_xmlencode (*data*)

XML encode unicode text.

Parameters *data* (*str*) – Unicode string to encode

Returns XML encoded data.

Return type *str*

class `azure.storage.queue.models.QueuePermissions` (*read=False, add=False, update=False, process=False, _str=None*)

Bases: `object`

QueuePermissions class to be used with `generate_queue_shared_access_signature()` method and for the AccessPolicies used with `set_queue_acl()`.

Variables

- `QueuePermissions.READ` (`QueuePermissions`) – Read metadata and properties, including message count. Peek at messages.
- `QueuePermissions.ADD` (`QueuePermissions`) – Add messages to the queue.
- `QueuePermissions.UPDATE` (`QueuePermissions`) – Update messages in the queue. Note: Use the Process permission with Update so you can first get the message you want to update.
- `QueuePermissions.PROCESS` (`QueuePermissions`) – Delete entities. Get and delete messages from the queue.

Parameters

- **read** (*bool*) – Read metadata and properties, including message count. Peek at messages.
- **add** (*bool*) – Add messages to the queue.
- **update** (*bool*) – Update messages in the queue. Note: Use the Process permission with Update so you can first get the message you want to update.
- **process** (*bool*) – Get and delete messages from the queue.
- **_str** (*str*) – A string representing the permissions.

ADD = <azure.storage.queue.models.QueuePermissions object>

PROCESS = <azure.storage.queue.models.QueuePermissions object>

READ = <azure.storage.queue.models.QueuePermissions object>

UPDATE = <azure.storage.queue.models.QueuePermissions object>

azure.storage.queue.queue.service module

class `azure.storage.queue.queue.service.QueueService` (*account_name=None, account_key=None, sas_token=None, is_emulated=False, protocol='https', endpoint_suffix='core.windows.net', request_session=None, connection_string=None, socket_timeout=None*)

Bases: `azure.storage.storageclient.StorageClient`

This is the main class managing queue resources.

The Queue service stores messages. A queue can contain an unlimited number of messages, each of which can be up to 64KB in size. Messages are generally added to the end of the queue and retrieved from the front of the queue, although first in, first out (FIFO) behavior is not guaranteed.

Variables

- **encode_function** (*function*(*data*)) – A function used to encode queue messages. Takes as a parameter the data passed to the `put_message` API and returns the encoded message. Defaults to take text and xml encode, but bytes and other encodings can be used. For example, base64 may be preferable for developing across multiple Azure Storage libraries in different languages. See the [QueueMessageFormat](#) for xml, base64 and no encoding methods as well as binary equivalents.
- **decode_function** (*function*(*data*)) – A function used to encode decode messages. Takes as a parameter the data returned by the `get_messages` and `peek_messages` APIs and returns the decoded message. Defaults to return text and xml decode, but bytes and other decodings can be used. For example, base64 may be preferable for developing across multiple Azure Storage libraries in different languages. See the [QueueMessageFormat](#) for xml, base64 and no decoding methods as well as binary equivalents.
- **key_encryption_key** (*object*) – The key-encryption-key optionally provided by the user. If provided, will be used to encrypt/decrypt in supported methods. For methods requiring decryption, either the `key_encryption_key` OR the resolver must be provided. If both are provided, the resolver will take precedence. Must implement the following methods for APIs requiring encryption: `wrap_key(key)`–wraps the specified key (bytes) using an algorithm of the user’s choice. Returns the encrypted key as bytes. `get_key_wrap_algorithm()`–returns the algorithm used to wrap the specified symmetric key. `get_kid()`–returns a string key id for this key-encryption-key. Must implement the following methods for APIs requiring decryption: `unwrap_key(key, algorithm)`–returns the unwrapped form of the specified symmetric key using the string-specified algorithm. `get_kid()`–returns a string key id for this key-encryption-key.
- **key_resolver_function** (*kid*) (*function*) – A function to resolve keys optionally provided by the user. If provided, will be used to decrypt in supported methods. For methods requiring decryption, either the `key_encryption_key` OR the resolver must be provided. If both are provided, the resolver will take precedence. It uses the `kid` string to return a key-encryption-key implementing the interface defined above.
- **require_encryption** (*bool*) – A flag that may be set to ensure that all messages successfully uploaded to the queue and all those downloaded and successfully read from the queue are/were encrypted while on the server. If this flag is set, all required parameters for encryption/decryption must be provided. See the above comments on the `key_encryption_key` and resolver.

Parameters

- **account_name** (*str*) – The storage account name. This is used to authenticate requests signed with an account key and to construct the storage endpoint. It is required unless a connection string is given.
- **account_key** (*str*) – The storage account key. This is used for shared key authentication.
- **sas_token** (*str*) – A shared access signature token to use to authenticate requests instead of the account key. If account key and sas token are both specified, account key will be used to sign.
- **is_emulated** (*bool*) – Whether to use the emulator. Defaults to False. If specified, will override all other parameters besides connection string and request session.

- **protocol** (*str*) – The protocol to use for requests. Defaults to https.
- **endpoint_suffix** (*str*) – The host base component of the url, minus the account name. Defaults to Azure (core.windows.net). Override this to use the China cloud (core.chinacloudapi.cn).
- **request_session** (*requests.Session*) – The session object to use for http requests.
- **connection_string** (*str*) – If specified, this will override all other parameters besides request session. See <http://azure.microsoft.com/en-us/documentation/articles/storage-configure-connection-string/> for the connection string format.
- **socket_timeout** (*int*) – If specified, this will override the default socket timeout. The timeout specified is in seconds. See DEFAULT_SOCKET_TIMEOUT in `_constants.py` for the default value.

clear_messages (*queue_name, timeout=None*)

Deletes all messages from the specified queue.

Parameters

- **queue_name** (*str*) – The name of the queue whose messages to clear.
- **timeout** (*int*) – The server timeout, expressed in seconds.

create_queue (*queue_name, metadata=None, fail_on_exist=False, timeout=None*)

Creates a queue under the given account.

Parameters

- **queue_name** (*str*) – The name of the queue to create. A queue name must be from 3 through 63 characters long and may only contain lowercase letters, numbers, and the dash (-) character. The first and last letters in the queue must be alphanumeric. The dash (-) character cannot be the first or last character. Consecutive dash characters are not permitted in the queue name.
- **metadata** (*a dict mapping str to str*) – A dict containing name-value pairs to associate with the queue as metadata. Note that metadata names preserve the case with which they were created, but are case-insensitive when set or read.
- **fail_on_exist** (*bool*) – Specifies whether to throw an exception if the queue already exists.
- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns A boolean indicating whether the queue was created. If `fail_on_exist` was set to `True`, this will throw instead of returning false.

Return type bool

delete_message (*queue_name, message_id, pop_receipt, timeout=None*)

Deletes the specified message.

Normally after a client retrieves a message with the `get_messages` operation, the client is expected to process and delete the message. To delete the message, you must have two items of data: `id` and `pop_receipt`. The `id` is returned from the previous `get_messages` operation. The `pop_receipt` is returned from the most recent `get_messages()` or `update_message()` operation. In order for the `delete_message` operation to succeed, the `pop_receipt` specified on the request must match the `pop_receipt` returned from the `get_messages()` or `update_message()` operation.

Parameters

- **queue_name** (*str*) – The name of the queue from which to delete the message.

- **message_id** (*str*) – The message id identifying the message to delete.
- **pop_receipt** (*str*) – A valid pop receipt value returned from an earlier call to the `get_messages()` or `update_message()`.
- **timeout** (*int*) – The server timeout, expressed in seconds.

delete_queue (*queue_name, fail_not_exist=False, timeout=None*)

Deletes the specified queue and any messages it contains.

When a queue is successfully deleted, it is immediately marked for deletion and is no longer accessible to clients. The queue is later removed from the Queue service during garbage collection.

Note that deleting a queue is likely to take at least 40 seconds to complete. If an operation is attempted against the queue while it was being deleted, an `AzureConflictHttpError` will be thrown.

Parameters

- **queue_name** (*str*) – The name of the queue to delete.
- **fail_not_exist** (*bool*) – Specifies whether to throw an exception if the queue doesn't exist.
- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns A boolean indicating whether the queue was deleted. If `fail_not_exist` was set to `True`, this will throw instead of returning `false`.

Return type `bool`

exists (*queue_name, timeout=None*)

Returns a boolean indicating whether the queue exists.

Parameters

- **queue_name** (*str*) – The name of queue to check for existence.
- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns A boolean indicating whether the queue exists.

Return type `bool`

generate_account_shared_access_signature (*resource_types, permission, expiry, start=None, ip=None, protocol=None*)

Generates a shared access signature for the queue service. Use the returned signature with the `sas_token` parameter of `QueueService`.

Parameters

- **resource_types** (`ResourceTypes`) – Specifies the resource types that are accessible with the account SAS.
- **permission** (`AccountPermissions`) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy.
- **expiry** (*datetime.datetime or str*) – The time at which the shared access signature becomes invalid. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.

- **start** (*datetime.datetime or str*) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **ip** (*str*) – Specifies an IP address or a range of IP addresses from which to accept requests. If the IP address from which the request originates does not match the IP address or address range specified on the SAS token, the request is not authenticated. For example, specifying `sip=168.1.5.65` or `sip=168.1.5.60-168.1.5.70` on the SAS restricts the request to those IP addresses.
- **protocol** (*str*) – Specifies the protocol permitted for a request made. The default value is `https,http`. See *Protocol* for possible values.

Returns A Shared Access Signature (sas) token.

Return type `str`

generate_queue_shared_access_signature (*queue_name, permission=None, expiry=None, start=None, id=None, ip=None, protocol=None*)

Generates a shared access signature for the queue. Use the returned signature with the `sas_token` parameter of `QueueService`.

Parameters

- **queue_name** (*str*) – The name of the queue to create a SAS token for.
- **permission** (`QueuePermissions`) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Required unless an `id` is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy.
- **expiry** (*datetime.datetime or str*) – The time at which the shared access signature becomes invalid. Required unless an `id` is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **start** (*datetime.datetime or str*) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **id** (*str*) – A unique value up to 64 characters in length that correlates to a stored access policy. To create a stored access policy, use `set_queue_acl()`.
- **ip** (*str*) – Specifies an IP address or a range of IP addresses from which to accept requests. If the IP address from which the request originates does not match the IP address or address range specified on the SAS token, the request is not authenticated. For example, specifying `sip='168.1.5.65'` or `sip='168.1.5.60-168.1.5.70'` on the SAS restricts the request to those IP addresses.
- **protocol** (*str*) – Specifies the protocol permitted for a request made. The default value is `https,http`. See *Protocol* for possible values.

Returns A Shared Access Signature (sas) token.

Return type `str`

get_messages (*queue_name, num_messages=None, visibility_timeout=None, timeout=None*)

Retrieves one or more messages from the front of the queue.

When a message is retrieved from the queue, the response includes the message content and a `pop_receipt` value, which is required to delete the message. The message is not automatically deleted from the queue, but after it has been retrieved, it is not visible to other clients for the time interval specified by the `visibility_timeout` parameter.

If the `key-encryption-key` or `resolver` field is set on the local service object, the messages will be decrypted before being returned.

Parameters

- **queue_name** (*str*) – The name of the queue to get messages from.
- **num_messages** (*int*) – A nonzero integer value that specifies the number of messages to retrieve from the queue, up to a maximum of 32. If fewer are visible, the visible messages are returned. By default, a single message is retrieved from the queue with this operation.
- **visibility_timeout** (*int*) – Specifies the new visibility timeout value, in seconds, relative to server time. The new value must be larger than or equal to 1 second, and cannot be larger than 7 days. The visibility timeout of a message can be set to a value later than the expiry time.
- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns A *QueueMessage* object representing the information passed.

Return type list of *QueueMessage*

get_queue_acl (*queue_name*, *timeout=None*)

Returns details about any stored access policies specified on the queue that may be used with Shared Access Signatures.

Parameters

- **queue_name** (*str*) – The name of an existing queue.
- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns A dictionary of access policies associated with the queue.

Return type dict of str to *AccessPolicy*

get_queue_metadata (*queue_name*, *timeout=None*)

Retrieves user-defined metadata and queue properties on the specified queue. Metadata is associated with the queue as name-value pairs.

Parameters

- **queue_name** (*str*) – The name of an existing queue.
- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns A dictionary representing the queue metadata with an `approximate_message_count` int property on the dict estimating the number of messages in the queue.

Return type a dict mapping str to str

get_queue_service_properties (*timeout=None*)

Gets the properties of a storage account's Queue service, including logging, analytics and CORS rules.

Parameters **timeout** (*int*) – The server timeout, expressed in seconds.

Returns The queue service properties.

Return type *ServiceProperties*

get_queue_service_stats (*timeout=None*)

Retrieves statistics related to replication for the Queue service. It is only available when read-access geo-redundant replication is enabled for the storage account.

With geo-redundant replication, Azure Storage maintains your data durable in two locations. In both locations, Azure Storage constantly maintains multiple healthy replicas of your data. The location where you read, create, update, or delete data is the primary storage account location. The primary location exists in the region you choose at the time you create an account via the Azure Management Azure classic portal, for example, North Central US. The location to which your data is replicated is the secondary location. The secondary location is automatically determined based on the location of the primary; it is in a second data center that resides in the same region as the primary location. Read-only access is available from the secondary location, if read-access geo-redundant replication is enabled for your storage account.

Parameters **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns The queue service stats.

Return type *ServiceStats*

list_queues (*prefix=None, num_results=None, include_metadata=False, marker=None, timeout=None*)

Returns a generator to list the queues. The generator will lazily follow the continuation tokens returned by the service and stop when all queues have been returned or `num_results` is reached.

If `num_results` is specified and the account has more than that number of queues, the generator will have a populated `next_marker` field once it finishes. This marker can be used to create a new generator if more results are desired.

Parameters

- **prefix** (*str*) – Filters the results to return only queues with names that begin with the specified prefix.
- **num_results** (*int*) – The maximum number of queues to return.
- **include_metadata** (*bool*) – Specifies that container metadata be returned in the response.
- **marker** (*str*) – An opaque continuation token. This value can be retrieved from the `next_marker` field of a previous generator object if `num_results` was specified and that generator has finished enumerating results. If specified, this generator will begin returning results from the point where the previous generator stopped.
- **timeout** (*int*) – The server timeout, expressed in seconds. This function may make multiple calls to the service in which case the timeout value specified will be applied to each individual call.

peek_messages (*queue_name, num_messages=None, timeout=None*)

Retrieves one or more messages from the front of the queue, but does not alter the visibility of the message.

Only messages that are visible may be retrieved. When a message is retrieved for the first time with a call to `get_messages`, its `dequeue_count` property is set to 1. If it is not deleted and is subsequently retrieved again, the `dequeue_count` property is incremented. The client may use this value to determine how many times a message has been retrieved. Note that a call to `peek_messages` does not increment the value of `DequeueCount`, but returns this value for the client to read.

If the `key-encryption-key` or `resolver` field is set on the local service object, the messages will be decrypted before being returned.

Parameters

- **queue_name** (*str*) – The name of the queue to peek messages from.

- **num_messages** (*int*) – A nonzero integer value that specifies the number of messages to peek from the queue, up to a maximum of 32. By default, a single message is peeked from the queue with this operation.
- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns A list of *QueueMessage* objects. Note that `time_next_visible` and `pop_receipt` will not be populated as `peek` does not pop the message and can only retrieve already visible messages.

Return type list of *QueueMessage*

put_message (*queue_name, content, visibility_timeout=None, time_to_live=None, timeout=None*)
 Adds a new message to the back of the message queue.

The visibility timeout specifies the time that the message will be invisible. After the timeout expires, the message will become visible. If a visibility timeout is not specified, the default value of 0 is used.

The message time-to-live specifies how long a message will remain in the queue. The message will be deleted from the queue when the time-to-live period expires.

If the key-encryption-key field is set on the local service object, this method will encrypt the content before uploading.

Parameters

- **queue_name** (*str*) – The name of the queue to put the message into.
- **content** (*obj*) – Message content. Allowed type is determined by the `encode_function` set on the service. Default is `str`. The encoded message can be up to 64KB in size.
- **visibility_timeout** (*int*) – If not specified, the default value is 0. Specifies the new visibility timeout value, in seconds, relative to server time. The value must be larger than or equal to 0, and cannot be larger than 7 days. The visibility timeout of a message cannot be set to a value later than the expiry time. `visibility_timeout` should be set to a value smaller than the time-to-live value.
- **time_to_live** (*int*) – Specifies the time-to-live interval for the message, in seconds. The maximum time-to-live allowed is 7 days. If this parameter is omitted, the default time-to-live is 7 days.
- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns A *QueueMessage* object. This object is also populated with the content although it is not returned from the service.

Return type *QueueMessage*

set_queue_acl (*queue_name, signed_identifiers=None, timeout=None*)
 Sets stored access policies for the queue that may be used with Shared Access Signatures.

When you set permissions for a queue, the existing permissions are replaced. To update the queue's permissions, call `get_queue_acl()` to fetch all access policies associated with the queue, modify the access policy that you wish to change, and then call this function with the complete set of data to perform the update.

When you establish a stored access policy on a queue, it may take up to 30 seconds to take effect. During this interval, a shared access signature that is associated with the stored access policy will throw an `AzureHttpError` until the access policy becomes active.

Parameters

- **queue_name** (*str*) – The name of an existing queue.

- **signed_identifiers** (dict of str to *AccessPolicy*) – A dictionary of access policies to associate with the queue. The dictionary may contain up to 5 elements. An empty dictionary will clear the access policies set on the service.
- **timeout** (*int*) – The server timeout, expressed in seconds.

set_queue_metadata (*queue_name, metadata=None, timeout=None*)

Sets user-defined metadata on the specified queue. Metadata is associated with the queue as name-value pairs.

Parameters

- **queue_name** (*str*) – The name of an existing queue.
- **metadata** (*dict*) – A dict containing name-value pairs to associate with the queue as metadata.
- **timeout** (*int*) – The server timeout, expressed in seconds.

set_queue_service_properties (*logging=None, hour_metrics=None, minute_metrics=None, cors=None, timeout=None*)

Sets the properties of a storage account's Queue service, including Azure Storage Analytics. If an element (ex Logging) is left as None, the existing settings on the service for that functionality are preserved. For more information on Azure Storage Analytics, see <https://msdn.microsoft.com/en-us/library/azure/hh343270.aspx>.

Parameters

- **logging** (*Logging*) – The logging settings provide request logs.
- **hour_metrics** (*Metrics*) – The hour metrics settings provide a summary of request statistics grouped by API in hourly aggregates for queues.
- **minute_metrics** (*Metrics*) – The minute metrics settings provide request statistics for each minute for queues.
- **cors** (list of *CorsRule*) – You can include up to five CorsRule elements in the list. If an empty list is specified, all CORS rules will be deleted, and CORS will be disabled for the service. For detailed information about CORS rules and evaluation logic, see <https://msdn.microsoft.com/en-us/library/azure/dn535601.aspx>.
- **timeout** (*int*) – The server timeout, expressed in seconds.

update_message (*queue_name, message_id, pop_receipt, visibility_timeout, content=None, timeout=None*)

Updates the visibility timeout of a message. You can also use this operation to update the contents of a message.

This operation can be used to continually extend the invisibility of a queue message. This functionality can be useful if you want a worker role to “lease” a queue message. For example, if a worker role calls `get_messages` and recognizes that it needs more time to process a message, it can continually extend the message's invisibility until it is processed. If the worker role were to fail during processing, eventually the message would become visible again and another worker role could process it.

If the key-encryption-key field is set on the local service object, this method will encrypt the content before uploading.

Parameters

- **queue_name** (*str*) – The name of the queue containing the message to update.
- **message_id** (*str*) – The message id identifying the message to update.

- **pop_receipt** (*str*) – A valid pop receipt value returned from an earlier call to the `get_messages()` or `update_message()` operation.
- **visibility_timeout** (*int*) – Specifies the new visibility timeout value, in seconds, relative to server time. The new value must be larger than or equal to 0, and cannot be larger than 7 days. The visibility timeout of a message cannot be set to a value later than the expiry time. A message can be updated until it has been deleted or has expired.
- **content** (*obj*) – Message content. Allowed type is determined by the `encode_function` set on the service. Default is `str`.
- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns A list of `QueueMessage` objects. For convenience, this object is also populated with the content, although it is not returned by the service.

Return type list of `QueueMessage`

Module contents

azure.storage.table package

Submodules

azure.storage.table.models module

exception `azure.storage.table.models.AzureBatchOperationError` (*message*, *status_code*, *batch_code*)

Bases: `azure.common.AzureHttpError`

Indicates that a batch operation failed.

Variables

- **message** (*str*) – A detailed error message indicating the index of the batch request which failed and the reason for the failure. For example, '0:One of the request inputs is out of range.' indicates the 0th batch request failed as one of its property values was out of range.
- **status_code** (*int*) – The HTTP status code of the batch request. For example, 400.
- **batch_code** (*str*) – The batch status code. For example, 'OutOfRangeInput'.

exception `azure.storage.table.models.AzureBatchValidationError`

Bases: `azure.common.AzureException`

Indicates that a batch operation cannot proceed due to invalid input.

Variables **message** (*str*) – A detailed error message indicating the reason for the failure.

class `azure.storage.table.models.EdmType`

Bases: `object`

Used by `EntityTypeProperty` to represent the type of the entity property to be stored by the Table service.

BINARY = 'Edm.Binary'

Represents byte data. Must be specified.

BOOLEAN = 'Edm.Boolean'

Represents a boolean. This type will be inferred for Python bools.

DATETIME = 'Edm.DateTime'

Represents a date. This type will be inferred for Python datetime objects.

DOUBLE = 'Edm.Double'

Represents a double. This type will be inferred for Python floating point numbers.

GUID = 'Edm.Guid'

Represents a GUID. Must be specified.

INT32 = 'Edm.Int32'

Represents a number between $-(2^{15})$ and 2^{15} . Must be specified or numbers will default to INT64.

INT64 = 'Edm.Int64'

Represents a number between $-(2^{31})$ and 2^{31} . This is the default type for Python numbers.

STRING = 'Edm.String'

Represents a string. This type will be inferred for Python strings.

class `azure.storage.table.models.Entity`

Bases: dict

An entity object. Can be accessed as a dict or as an obj. The attributes of the entity will be created dynamically. For example, the following are both valid:

```
entity = Entity()
entity.a = 'b'
entity['x'] = 'y'
```

class `azure.storage.table.models.EntityProperty` (*type=None, value=None, encrypt=False*)

Bases: object

An entity property. Used to explicitly set *EdmType* when necessary.

Values which require explicit typing are GUID, INT32, and BINARY. Other EdmTypes may be explicitly create as EntityProperty objects but need not be. For example, the below with both create STRING typed properties on the entity:

```
entity = Entity()
entity.a = 'b'
entity.x = EntityProperty(EdmType.STRING, 'y')
```

Represents an Azure Table. Returned by `list_tables`.

Parameters

- **type** (*str*) – The type of the property.
- **value** (*EdmType*) – The value of the property.
- **encrypt** (*bool*) – Indicates whether or not the property should be encrypted.

class `azure.storage.table.models.Table`

Bases: object

Represents an Azure Table. Returned by `list_tables`.

Variables `name` (*str*) – The name of the table.

class `azure.storage.table.models.TablePayloadFormat`

Bases: object

Specifies the accepted content type of the response payload. More information can be found here: <https://msdn.microsoft.com/en-us/library/azure/dn535600.aspx>

JSON_FULL_METADATA = 'application/json;odata=fullmetadata'

Returns minimal type information for the entity properties plus some extra odata properties.

JSON_MINIMAL_METADATA = 'application/json;odata=minimalmetadata'

Returns minimal type information for the entity properties.

JSON_NO_METADATA = 'application/json;odata=nometadata'

Returns no type information for the entity properties.

class azure.storage.table.models.**TablePermissions** (*query=False, add=False, update=False, delete=False, _str=None*)

Bases: object

TablePermissions class to be used with the `generate_table_shared_access_signature()` method and for the AccessPolicies used with `set_table_acl()`.

Variables

- **TablePermissions.QUERY** (TablePermissions) – Get entities and query entities.
- **TablePermissions.ADD** (TablePermissions) – Add entities.
- **TablePermissions.UPDATE** (TablePermissions) – Update entities.
- **TablePermissions.DELETE** (TablePermissions) – Delete entities.

Parameters

- **query** (*bool*) – Get entities and query entities.
- **add** (*bool*) – Add entities. Add and Update permissions are required for upsert operations.
- **update** (*bool*) – Update entities. Add and Update permissions are required for upsert operations.
- **delete** (*bool*) – Delete entities.
- **_str** (*str*) – A string representing the permissions.

ADD = <azure.storage.table.models.TablePermissions object>

DELETE = <azure.storage.table.models.TablePermissions object>

QUERY = <azure.storage.table.models.TablePermissions object>

UPDATE = <azure.storage.table.models.TablePermissions object>

azure.storage.table.tablebatch module

class azure.storage.table.tablebatch.**TableBatch** (*require_encryption=False, key_encryption_key=None, encryption_resolver=None*)

Bases: object

This is the class that is used for batch operation for storage table service.

The Table service supports batch transactions on entities that are in the same table and belong to the same partition group. Multiple operations are supported within a single transaction. The batch can include at most 100 entities, and its total payload may be no more than 4 MB in size.

delete_entity (*partition_key, row_key, if_match='*'*)

Adds a delete entity operation to the batch. See `delete_entity()` for more information on deletes.

The operation will not be executed until the batch is committed.

Parameters

- **partition_key** (*str*) – The PartitionKey of the entity.
- **row_key** (*str*) – The RowKey of the entity.
- **if_match** (*str*) – The client may specify the ETag for the entity on the request in order to compare to the ETag maintained by the service for the purpose of optimistic concurrency. The delete operation will be performed only if the ETag sent by the client matches the value maintained by the server, indicating that the entity has not been modified since it was retrieved by the client. To force an unconditional delete, set If-Match to the wildcard character (*).

insert_entity (*entity*)

Adds an insert entity operation to the batch. See `insert_entity()` for more information on inserts.

The operation will not be executed until the batch is committed.

Parameters **entity** (a dict or `azure.storage.table.models.Entity`) – The entity to insert. Could be a dict or an entity object. Must contain a PartitionKey and a RowKey.

insert_or_merge_entity (*entity*)

Adds an insert or merge entity operation to the batch. See `insert_or_merge_entity()` for more information on insert or merge operations.

The operation will not be executed until the batch is committed.

Parameters **entity** (a dict or `azure.storage.table.models.Entity`) – The entity to insert or merge. Could be a dict or an entity object. Must contain a PartitionKey and a RowKey.

insert_or_replace_entity (*entity*)

Adds an insert or replace entity operation to the batch. See `insert_or_replace_entity()` for more information on insert or replace operations.

The operation will not be executed until the batch is committed.

Parameters **entity** (a dict or `azure.storage.table.models.Entity`) – The entity to insert or replace. Could be a dict or an entity object. Must contain a PartitionKey and a RowKey.

merge_entity (*entity*, *if_match='*'*)

Adds a merge entity operation to the batch. See `merge_entity()` for more information on merges.

The operation will not be executed until the batch is committed.

Parameters

- **entity** (a dict or `azure.storage.table.models.Entity`) – The entity to merge. Could be a dict or an entity object. Must contain a PartitionKey and a RowKey.
- **if_match** (*str*) – The client may specify the ETag for the entity on the request in order to compare to the ETag maintained by the service for the purpose of optimistic concurrency. The merge operation will be performed only if the ETag sent by the client matches the value maintained by the server, indicating that the entity has not been modified since it was retrieved by the client. To force an unconditional merge, set If-Match to the wildcard character (*).

update_entity (*entity*, *if_match='*'*)

Adds an update entity operation to the batch. See `update_entity()` for more information on updates.

The operation will not be executed until the batch is committed.

Parameters

- **entity** (a dict or `azure.storage.table.models.Entity`) – The entity to update. Could be a dict or an entity object. Must contain a PartitionKey and a RowKey.
- **if_match** (`str`) – The client may specify the ETag for the entity on the request in order to compare to the ETag maintained by the service for the purpose of optimistic concurrency. The update operation will be performed only if the ETag sent by the client matches the value maintained by the server, indicating that the entity has not been modified since it was retrieved by the client. To force an unconditional update, set If-Match to the wildcard character (*).

azure.storage.table.tableservice module

```
class azure.storage.table.tableservice.TableService (account_name=None,          ac-
                                                    count_key=None, sas_token=None,
                                                    is_emulated=False,          pro-
                                                    tocol='https',              end-
                                                    point_suffix='core.windows.net',
                                                    request_session=None,
                                                    connection_string=None,
                                                    socket_timeout=None)
```

Bases: `azure.storage.storageclient.StorageClient`

This is the main class managing Azure Table resources.

The Azure Table service offers structured storage in the form of tables. Tables store data as collections of entities. Entities are similar to rows. An entity has a primary key and a set of properties. A property is a name, typed-value pair, similar to a column. The Table service does not enforce any schema for tables, so two entities in the same table may have different sets of properties. Developers may choose to enforce a schema on the client side. A table may contain any number of entities.

Variables

- **key_encryption_key** (`object`) – The key-encryption-key optionally provided by the user. If provided, will be used to encrypt/decrypt in supported methods. For methods requiring decryption, either the `key_encryption_key` OR the resolver must be provided. If both are provided, the resolver will take precedence. Must implement the following methods for APIs requiring encryption: `wrap_key(key)`–wraps the specified key (bytes) using an algorithm of the user’s choice. Returns the encrypted key as bytes. `get_key_wrap_algorithm()`–returns the algorithm used to wrap the specified symmetric key. `get_kid()`–returns a string key id for this key-encryption-key. Must implement the following methods for APIs requiring decryption: `unwrap_key(key, algorithm)`–returns the unwrapped form of the specified symmetric key using the string-specified algorithm. `get_kid()`–returns a string key id for this key-encryption-key.
- **key_resolver_function(kid)** (`function`) – A function to resolve keys optionally provided by the user. If provided, will be used to decrypt in supported methods. For methods requiring decryption, either the `key_encryption_key` OR the resolver must be provided. If both are provided, the resolver will take precedence. It uses the kid string to return a key-encryption-key implementing the interface defined above.
- **row_key, property_name) encryption_resolver_functions** (`function(partition_key,)`) – A function that takes in an entity’s partition key, row key, and property name and returns a boolean that indicates whether that property should be encrypted.
- **require_encryption** (`bool`) – A flag that may be set to ensure that all messages successfully uploaded to the queue and all those downloaded and successfully read from

the queue are/were encrypted while on the server. If this flag is set, all required parameters for encryption/decryption must be provided. See the above comments on the `key_encryption_key` and `resolver`.

Parameters

- **account_name** (*str*) – The storage account name. This is used to authenticate requests signed with an account key and to construct the storage endpoint. It is required unless a connection string is given.
- **account_key** (*str*) – The storage account key. This is used for shared key authentication.
- **sas_token** (*str*) – A shared access signature token to use to authenticate requests instead of the account key. If account key and sas token are both specified, account key will be used to sign.
- **is_emulated** (*bool*) – Whether to use the emulator. Defaults to False. If specified, will override all other parameters besides connection string and request session.
- **protocol** (*str*) – The protocol to use for requests. Defaults to https.
- **endpoint_suffix** (*str*) – The host base component of the url, minus the account name. Defaults to Azure (core.windows.net). Override this to use the China cloud (core.chinacloudapi.cn).
- **request_session** (*requests.Session*) – The session object to use for http requests.
- **connection_string** (*str*) – If specified, this will override all other parameters besides request session. See <http://azure.microsoft.com/en-us/documentation/articles/storage-configure-connection-string/> for the connection string format.
- **socket_timeout** (*int*) – If specified, this will override the default socket timeout. The timeout specified is in seconds. See `DEFAULT_SOCKET_TIMEOUT` in `_constants.py` for the default value.

batch (*args, **kws)

Creates a batch object which can be used as a context manager. Commits the batch on exit.

Parameters

- **table_name** (*str*) – The name of the table to commit the batch to.
- **timeout** (*int*) – The server timeout, expressed in seconds.

commit_batch (*table_name, batch, timeout=None*)

Commits a `TableBatch` request.

Parameters

- **table_name** (*str*) – The name of the table to commit the batch to.
- **batch** (`TableBatch`) – The batch to commit.
- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns A list of the batch responses corresponding to the requests in the batch.

Return type list of response objects

create_table (*table_name, fail_on_exist=False, timeout=None*)

Creates a new table in the storage account.

Parameters

- **table_name** (*str*) – The name of the table to create. The table name may contain only alphanumeric characters and cannot begin with a numeric character. It is case-insensitive and must be from 3 to 63 characters long.
- **fail_on_exist** (*bool*) – Specifies whether to throw an exception if the table already exists.
- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns A boolean indicating whether the table was created. If `fail_on_exist` was set to `True`, this will throw instead of returning `false`.

Return type `bool`

delete_entity (*table_name, partition_key, row_key, if_match='*', timeout=None*)

Deletes an existing entity in a table. Throws if the entity does not exist.

When an entity is successfully deleted, the entity is immediately marked for deletion and is no longer accessible to clients. The entity is later removed from the Table service during garbage collection.

Parameters

- **table_name** (*str*) – The name of the table containing the entity to delete.
- **partition_key** (*str*) – The `PartitionKey` of the entity.
- **row_key** (*str*) – The `RowKey` of the entity.
- **if_match** (*str*) – The client may specify the ETag for the entity on the request in order to compare to the ETag maintained by the service for the purpose of optimistic concurrency. The delete operation will be performed only if the ETag sent by the client matches the value maintained by the server, indicating that the entity has not been modified since it was retrieved by the client. To force an unconditional delete, set `If-Match` to the wildcard character (*).
- **timeout** (*int*) – The server timeout, expressed in seconds.

delete_table (*table_name, fail_not_exist=False, timeout=None*)

Deletes the specified table and any data it contains.

When a table is successfully deleted, it is immediately marked for deletion and is no longer accessible to clients. The table is later removed from the Table service during garbage collection.

Note that deleting a table is likely to take at least 40 seconds to complete. If an operation is attempted against the table while it was being deleted, an `AzureConflictHttpError` will be thrown.

Parameters

- **table_name** (*str*) – The name of the table to delete.
- **fail_not_exist** (*bool*) – Specifies whether to throw an exception if the table doesn't exist.
- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns A boolean indicating whether the table was deleted. If `fail_not_exist` was set to `True`, this will throw instead of returning `false`.

Return type `bool`

exists (*table_name, timeout=None*)

Returns a boolean indicating whether the table exists.

Parameters

- **table_name** (*str*) – The name of table to check for existence.

- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns A boolean indicating whether the table exists.

Return type bool

generate_account_shared_access_signature (*resource_types, permission, expiry, start=None, ip=None, protocol=None*)

Generates a shared access signature for the table service. Use the returned signature with the `sas_token` parameter of `TableService`.

Parameters

- **resource_types** (`ResourceTypes`) – Specifies the resource types that are accessible with the account SAS.
- **permission** (`AccountPermissions`) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Required unless an `id` is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy.
- **expiry** (*datetime.datetime or str*) – The time at which the shared access signature becomes invalid. Required unless an `id` is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **start** (*datetime.datetime or str*) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **ip** (*str*) – Specifies an IP address or a range of IP addresses from which to accept requests. If the IP address from which the request originates does not match the IP address or address range specified on the SAS token, the request is not authenticated. For example, specifying `sip=168.1.5.65` or `sip=168.1.5.60-168.1.5.70` on the SAS restricts the request to those IP addresses.
- **protocol** (*str*) – Specifies the protocol permitted for a request made. The default value is `https,http`. See `Protocol` for possible values.

Returns A Shared Access Signature (sas) token.

Return type str

generate_table_shared_access_signature (*table_name, permission=None, expiry=None, start=None, id=None, ip=None, protocol=None, start_pk=None, start_rk=None, end_pk=None, end_rk=None*)

Generates a shared access signature for the table. Use the returned signature with the `sas_token` parameter of `TableService`.

Parameters

- **table_name** (*str*) – The name of the table to create a SAS token for.
- **permission** (`TablePermissions`) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Required unless an `id` is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy.
- **expiry** (*datetime.datetime or str*) – The time at which the shared access signature becomes invalid. Required unless an `id` is given referencing a stored access policy

which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.

- **start** (*datetime.datetime* or *str*) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **id** (*str*) – A unique value up to 64 characters in length that correlates to a stored access policy. To create a stored access policy, use `set_table_acl()`.
- **ip** (*str*) – Specifies an IP address or a range of IP addresses from which to accept requests. If the IP address from which the request originates does not match the IP address or address range specified on the SAS token, the request is not authenticated. For example, specifying `sip='168.1.5.65'` or `sip='168.1.5.60-168.1.5.70'` on the SAS restricts the request to those IP addresses.
- **protocol** (*str*) – Specifies the protocol permitted for a request made. The default value is `https,http`. See *Protocol* for possible values.
- **start_pk** (*str*) – The minimum partition key accessible with this shared access signature. `startpk` must accompany `startrk`. Key values are inclusive. If omitted, there is no lower bound on the table entities that can be accessed.
- **start_rk** (*str*) – The minimum row key accessible with this shared access signature. `startpk` must accompany `startrk`. Key values are inclusive. If omitted, there is no lower bound on the table entities that can be accessed.
- **end_pk** (*str*) – The maximum partition key accessible with this shared access signature. `endpk` must accompany `endrk`. Key values are inclusive. If omitted, there is no upper bound on the table entities that can be accessed.
- **end_rk** (*str*) – The maximum row key accessible with this shared access signature. `endpk` must accompany `endrk`. Key values are inclusive. If omitted, there is no upper bound on the table entities that can be accessed.

Returns A Shared Access Signature (sas) token.

Return type `str`

get_entity (*table_name*, *partition_key*, *row_key*, *select=None*, *accept='application/json;odata=minimalmetadata'*, *property_resolver=None*, *timeout=None*)

Get an entity from the specified table. Throws if the entity does not exist.

Parameters

- **table_name** (*str*) – The name of the table to get the entity from.
- **partition_key** (*str*) – The PartitionKey of the entity.
- **row_key** (*str*) – The RowKey of the entity.
- **select** (*str*) – Returns only the desired properties of an entity from the set.
- **accept** (*str*) – Specifies the accepted content type of the response payload. See *TablePayloadFormat* for possible values.
- **property_resolver** (*callback function in format of func(pk, rk, prop_name, prop_value, service_edm_type)*) – A function which given the partition key, row key, property name, property value, and the property EdmType

if returned by the service, returns the EdmType of the property. Generally used if `accept` is set to `JSON_NO_METADATA`.

- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns The retrieved entity.

Return type *Entity*

get_table_acl (*table_name*, *timeout=None*)

Returns details about any stored access policies specified on the table that may be used with Shared Access Signatures.

Parameters

- **table_name** (*str*) – The name of an existing table.
- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns A dictionary of access policies associated with the table.

Return type dict of str to *AccessPolicy*:

get_table_service_properties (*timeout=None*)

Gets the properties of a storage account's Table service, including logging, analytics and CORS rules.

Parameters **timeout** (*int*) – The server timeout, expressed in seconds.

Returns The table service properties.

Return type *ServiceProperties*

get_table_service_stats (*timeout=None*)

Retrieves statistics related to replication for the Table service. It is only available when read-access geo-redundant replication is enabled for the storage account.

With geo-redundant replication, Azure Storage maintains your data durable in two locations. In both locations, Azure Storage constantly maintains multiple healthy replicas of your data. The location where you read, create, update, or delete data is the primary storage account location. The primary location exists in the region you choose at the time you create an account via the Azure Management Azure classic portal, for example, North Central US. The location to which your data is replicated is the secondary location. The secondary location is automatically determined based on the location of the primary; it is in a second data center that resides in the same region as the primary location. Read-only access is available from the secondary location, if read-access geo-redundant replication is enabled for your storage account.

Parameters **timeout** (*int*) – The timeout parameter is expressed in seconds.

Returns The table service stats.

Return type *ServiceStats*

insert_entity (*table_name*, *entity*, *timeout=None*)

Inserts a new entity into the table. Throws if an entity with the same PartitionKey and RowKey already exists.

When inserting an entity into a table, you must specify values for the PartitionKey and RowKey system properties. Together, these properties form the primary key and must be unique within the table. Both the PartitionKey and RowKey values must be string values; each key value may be up to 64 KB in size. If you are using an integer value for the key value, you should convert the integer to a fixed-width string, because they are canonically sorted. For example, you should convert the value 1 to 0000001 to ensure proper sorting.

Parameters

- **table_name** (*str*) – The name of the table to insert the entity into.

- **entity** (a dict or *Entity*) – The entity to insert. Could be a dict or an entity object. Must contain a PartitionKey and a RowKey.
- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns The etag of the inserted entity.

Return type str

insert_or_merge_entity (*table_name*, *entity*, *timeout=None*)

Merges an existing entity or inserts a new entity if it does not exist in the table.

If `insert_or_merge_entity` is used to merge an entity, any properties from the previous entity will be retained if the request does not define or include them.

Parameters

- **table_name** (*str*) – The name of the table in which to insert or merge the entity.
- **entity** (a dict or *Entity*) – The entity to insert or merge. Could be a dict or an entity object. Must contain a PartitionKey and a RowKey.
- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns The etag of the entity.

Return type str

insert_or_replace_entity (*table_name*, *entity*, *timeout=None*)

Replaces an existing entity or inserts a new entity if it does not exist in the table. Because this operation can insert or update an entity, it is also known as an “upsert” operation.

If `insert_or_replace_entity` is used to replace an entity, any properties from the previous entity will be removed if the new entity does not define them.

Parameters

- **table_name** (*str*) – The name of the table in which to insert or replace the entity.
- **entity** (a dict or *Entity*) – The entity to insert or replace. Could be a dict or an entity object. Must contain a PartitionKey and a RowKey.
- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns The etag of the entity.

Return type str

list_tables (*num_results=None*, *marker=None*, *timeout=None*)

Returns a generator to list the tables. The generator will lazily follow the continuation tokens returned by the service and stop when all tables have been returned or `num_results` is reached.

If `num_results` is specified and the account has more than that number of tables, the generator will have a populated `next_marker` field once it finishes. This marker can be used to create a new generator if more results are desired.

Parameters

- **num_results** (*int*) – The maximum number of tables to return.
- **marker** (*obj*) – An opaque continuation object. This value can be retrieved from the `next_marker` field of a previous generator object if `num_results` was specified and that generator has finished enumerating results. If specified, this generator will begin returning results from the point where the previous generator stopped.

- **timeout** (*int*) – The server timeout, expressed in seconds. This function may make multiple calls to the service in which case the timeout value specified will be applied to each individual call.

Returns A generator which produces `Table` objects.

Return type `ListGenerator`:

merge_entity (*table_name*, *entity*, *if_match='*'*, *timeout=None*)

Updates an existing entity by merging the entity's properties. Throws if the entity does not exist.

This operation does not replace the existing entity as the `update_entity` operation does. A property cannot be removed with `merge_entity`.

Any properties with null values are ignored. All other properties will be updated or added.

Parameters

- **table_name** (*str*) – The name of the table containing the entity to merge.
- **entity** (a dict or `Entity`) – The entity to merge. Could be a dict or an entity object. Must contain a `PartitionKey` and a `RowKey`.
- **if_match** (*str*) – The client may specify the ETag for the entity on the request in order to compare to the ETag maintained by the service for the purpose of optimistic concurrency. The merge operation will be performed only if the ETag sent by the client matches the value maintained by the server, indicating that the entity has not been modified since it was retrieved by the client. To force an unconditional merge, set If-Match to the wildcard character (*).
- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns The etag of the entity.

Return type `str`

query_entities (*table_name*, *filter=None*, *select=None*, *num_results=None*, *marker=None*, *accept='application/json;odata=minimalmetadata'*, *property_resolver=None*, *timeout=None*)

Returns a generator to list the entities in the table specified. The generator will lazily follow the continuation tokens returned by the service and stop when all entities have been returned or `max_results` is reached.

If `max_results` is specified and the account has more than that number of entities, the generator will have a populated `next_marker` field once it finishes. This marker can be used to create a new generator if more results are desired.

Parameters

- **table_name** (*str*) – The name of the table to query.
- **filter** (*str*) – Returns only entities that satisfy the specified filter. Note that no more than 15 discrete comparisons are permitted within a `$filter` string. See <http://msdn.microsoft.com/en-us/library/windowsazure/dd894031.aspx> for more information on constructing filters.
- **select** (*str*) – Returns only the desired properties of an entity from the set.
- **num_results** (*int*) – The maximum number of entities to return.
- **marker** (*obj*) – An opaque continuation object. This value can be retrieved from the `next_marker` field of a previous generator object if `max_results` was specified and that generator has finished enumerating results. If specified, this generator will begin returning results from the point where the previous generator stopped.

- **accept** (*str*) – Specifies the accepted content type of the response payload. See *TablePayloadFormat* for possible values.
- **property_resolver** (*callback function in format of func(pk, rk, prop_name, prop_value, service_edm_type)*) – A function which given the partition key, row key, property name, property value, and the property EdmType if returned by the service, returns the EdmType of the property. Generally used if `accept` is set to `JSON_NO_METADATA`.
- **timeout** (*int*) – The server timeout, expressed in seconds. This function may make multiple calls to the service in which case the timeout value specified will be applied to each individual call.

Returns A generator which produces *Entity* objects.

Return type *ListGenerator*

set_table_acl (*table_name, signed_identifiers=None, timeout=None*)

Sets stored access policies for the table that may be used with Shared Access Signatures.

When you set permissions for a table, the existing permissions are replaced. To update the table's permissions, call *get_table_acl()* to fetch all access policies associated with the table, modify the access policy that you wish to change, and then call this function with the complete set of data to perform the update.

When you establish a stored access policy on a table, it may take up to 30 seconds to take effect. During this interval, a shared access signature that is associated with the stored access policy will throw an *AzureHttpError* until the access policy becomes active.

Parameters

- **table_name** (*str*) – The name of an existing table.
- **signed_identifiers** (dict of *str* to *AccessPolicy*) – A dictionary of access policies to associate with the table. The dictionary may contain up to 5 elements. An empty dictionary will clear the access policies set on the service.
- **timeout** (*int*) – The server timeout, expressed in seconds.

set_table_service_properties (*logging=None, hour_metrics=None, minute_metrics=None, cors=None, timeout=None*)

Sets the properties of a storage account's Table service, including Azure Storage Analytics. If an element (ex Logging) is left as *None*, the existing settings on the service for that functionality are preserved. For more information on Azure Storage Analytics, see <https://msdn.microsoft.com/en-us/library/azure/hh343270.aspx>.

Parameters

- **logging** (*Logging*) – The logging settings provide request logs.
- **hour_metrics** (*Metrics*) – The hour metrics settings provide a summary of request statistics grouped by API in hourly aggregates for tables.
- **minute_metrics** (*Metrics*) – The minute metrics settings provide request statistics for each minute for tables.
- **cors** (list of *CorsRule*) – You can include up to five *CorsRule* elements in the list. If an empty list is specified, all CORS rules will be deleted, and CORS will be disabled for the service. For detailed information about CORS rules and evaluation logic, see <https://msdn.microsoft.com/en-us/library/azure/dn535601.aspx>.
- **timeout** (*int*) – The server timeout, expressed in seconds.

update_entity (*table_name*, *entity*, *if_match='*'*, *timeout=None*)

Updates an existing entity in a table. Throws if the entity does not exist. The update_entity operation replaces the entire entity and can be used to remove properties.

Parameters

- **table_name** (*str*) – The name of the table containing the entity to update.
- **entity** (a dict or *Entity*) – The entity to update. Could be a dict or an entity object. Must contain a PartitionKey and a RowKey.
- **if_match** (*str*) – The client may specify the ETag for the entity on the request in order to compare to the ETag maintained by the service for the purpose of optimistic concurrency. The update operation will be performed only if the ETag sent by the client matches the value maintained by the server, indicating that the entity has not been modified since it was retrieved by the client. To force an unconditional update, set If-Match to the wildcard character (*).
- **timeout** (*int*) – The server timeout, expressed in seconds.

Returns The etag of the entity.

Return type *str*

Module contents

Submodules

azure.storage.cloudstorageaccount module

class `azure.storage.cloudstorageaccount.CloudStorageAccount` (*account_name=None*,
account_key=None,
sas_token=None,
is_emulated=None)

Bases: `object`

Provides a factory for creating the blob, queue, table, and file services with a common account name and account key or sas token. Users can either use the factory or can construct the appropriate service directly.

Parameters

- **account_name** (*str*) – The storage account name. This is used to authenticate requests signed with an account key and to construct the storage endpoint. It is required unless *is_emulated* is used.
- **account_key** (*str*) – The storage account key. This is used for shared key authentication.
- **sas_token** (*str*) – A shared access signature token to use to authenticate requests instead of the account key. If account key and sas token are both specified, account key will be used to sign.
- **is_emulated** (*bool*) – Whether to use the emulator. Defaults to False. If specified, will override all other parameters.

create_append_blob_service ()

Creates a AppendBlobService object with the settings specified in the CloudStorageAccount.

Returns A service object.

Return type *AppendBlobService*

create_block_blob_service()

Creates a BlockBlobService object with the settings specified in the CloudStorageAccount.

Returns A service object.

Return type *BlockBlobService*

create_file_service()

Creates a FileService object with the settings specified in the CloudStorageAccount.

Returns A service object.

Return type *FileService*

create_page_blob_service()

Creates a PageBlobService object with the settings specified in the CloudStorageAccount.

Returns A service object.

Return type *PageBlobService*

create_queue_service()

Creates a QueueService object with the settings specified in the CloudStorageAccount.

Returns A service object.

Return type *QueueService*

create_table_service()

Creates a TableService object with the settings specified in the CloudStorageAccount.

Returns A service object.

Return type *TableService*

generate_shared_access_signature (*services, resource_types, permission, expiry, start=None, ip=None, protocol=None*)

Generates a shared access signature for the account. Use the returned signature with the sas_token parameter of the service or to create a new account object.

Parameters

- **services** (*Services*) – Specifies the services accessible with the account SAS. You can combine values to provide access to more than one service.
- **resource_types** (*ResourceTypes*) – Specifies the resource types that are accessible with the account SAS. You can combine values to provide access to more than one resource type.
- **permission** (*AccountPermissions*) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. You can combine values to provide more than one permission.
- **expiry** (*datetime.datetime or str*) – The time at which the shared access signature becomes invalid. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **start** (*datetime.datetime or str*) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.

- **ip** (*str*) – Specifies an IP address or a range of IP addresses from which to accept requests. If the IP address from which the request originates does not match the IP address or address range specified on the SAS token, the request is not authenticated. For example, specifying `sip=168.1.5.65` or `sip=168.1.5.60-168.1.5.70` on the SAS restricts the request to those IP addresses.
- **protocol** (*str*) – Specifies the protocol permitted for a request made. Possible values are both HTTPS and HTTP (`https,http`) or HTTPS only (`https`). The default value is `https,http`. Note that HTTP only is not a permitted value.

azure.storage.models module

class `azure.storage.models.AccessPolicy` (*permission=None, expiry=None, start=None*)

Bases: `object`

Access Policy class used by the set and get acl methods in each service.

A stored access policy can specify the start time, expiry time, and permissions for the Shared Access Signatures with which it's associated. Depending on how you want to control access to your table resource, you can specify all of these parameters within the stored access policy, and omit them from the URL for the Shared Access Signature. Doing so permits you to modify the associated signature's behavior at any time, as well as to revoke it. Or you can specify one or more of the access policy parameters within the stored access policy, and the others on the URL. Finally, you can specify all of the parameters on the URL. In this case, you can use the stored access policy to revoke the signature, but not to modify its behavior.

Together the Shared Access Signature and the stored access policy must include all fields required to authenticate the signature. If any required fields are missing, the request will fail. Likewise, if a field is specified both in the Shared Access Signature URL and in the stored access policy, the request will fail with status code 400 (Bad Request).

Parameters

- **permission** (*str*) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy.
- **expiry** (*datetime.datetime or str*) – The time at which the shared access signature becomes invalid. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **start** (*datetime.datetime or str*) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.

class `azure.storage.models.AccountPermissions` (*read=False, write=False, delete=False, list=False, add=False, create=False, update=False, process=False, _str=None*)

Bases: `object`

ResourceTypes class to be used with `generate_shared_access_signature` method and for the `AccessPolicies` used with `set_*_acl`. There are two types of SAS which may be used to grant resource access. One is to grant access to a specific resource (resource-specific). Another is to grant access to the entire service for a specific account and allow certain operations based on perms found here.

Variables

- **`AccountPermissions.ADD`** (`AccountPermissions`) – Valid for the following Object resource types only: queue messages, table entities, and append blobs.
- **`AccountPermissions.CREATE`** (`AccountPermissions`) – Valid for the following Object resource types only: blobs and files. Users can create new blobs or files, but may not overwrite existing blobs or files.
- **`AccountPermissions.DELETE`** (`AccountPermissions`) – Valid for Container and Object resource types, except for queue messages.
- **`AccountPermissions.LIST`** (`AccountPermissions`) – Valid for Service and Container resource types only.
- **`AccountPermissions.PROCESS`** (`AccountPermissions`) – Valid for the following Object resource type only: queue messages.
- **`AccountPermissions.READ`** (`AccountPermissions`) – Valid for all signed resources types (Service, Container, and Object). Permits read permissions to the specified resource type.
- **`AccountPermissions.UPDATE`** (`AccountPermissions`) – Valid for the following Object resource types only: queue messages and table entities.
- **`AccountPermissions.WRITE`** (`AccountPermissions`) – Valid for all signed resources types (Service, Container, and Object). Permits write permissions to the specified resource type.

Parameters

- **`read`** (`bool`) – Valid for all signed resources types (Service, Container, and Object). Permits read permissions to the specified resource type.
- **`write`** (`bool`) – Valid for all signed resources types (Service, Container, and Object). Permits write permissions to the specified resource type.
- **`delete`** (`bool`) – Valid for Container and Object resource types, except for queue messages.
- **`list`** (`bool`) – Valid for Service and Container resource types only.
- **`add`** (`bool`) – Valid for the following Object resource types only: queue messages, table entities, and append blobs.
- **`create`** (`bool`) – Valid for the following Object resource types only: blobs and files. Users can create new blobs or files, but may not overwrite existing blobs or files.
- **`update`** (`bool`) – Valid for the following Object resource types only: queue messages and table entities.
- **`process`** (`bool`) – Valid for the following Object resource type only: queue messages.
- **`_str`** (`str`) – A string representing the permissions.

ADD = <azure.storage.models.AccountPermissions object>

CREATE = <azure.storage.models.AccountPermissions object>

DELETE = <azure.storage.models.AccountPermissions object>

LIST = <azure.storage.models.AccountPermissions object>

PROCESS = <azure.storage.models.AccountPermissions object>

READ = <azure.storage.models.AccountPermissions object>

UPDATE = <azure.storage.models.AccountPermissions object>

WRITE = <azure.storage.models.AccountPermissions object>

```
class azure.storage.models.CorsRule(allowed_origins, allowed_methods,
                                     max_age_in_seconds=0, exposed_headers=None,
                                     allowed_headers=None)
```

Bases: object

CORS is an HTTP feature that enables a web application running under one domain to access resources in another domain. Web browsers implement a security restriction known as same-origin policy that prevents a web page from calling APIs in a different domain; CORS provides a secure way to allow one domain (the origin domain) to call APIs in another domain.

For more information, see <https://msdn.microsoft.com/en-us/library/azure/dn535601.aspx>

Parameters

- **allowed_origins** (*list of str*) – A list of origin domains that will be allowed via CORS, or "*" to allow all domains. The list of must contain at least one entry. Limited to 64 origin domains. Each allowed origin can have up to 256 characters.
- **allowed_methods** (*list of str*) – A list of HTTP methods that are allowed to be executed by the origin. The list of must contain at least one entry. For Azure Storage, permitted methods are DELETE, GET, HEAD, MERGE, POST, OPTIONS or PUT.
- **max_age_in_seconds** (*int*) – The number of seconds that the client/browser should cache a preflight response.
- **exposed_headers** (*list of str*) – Defaults to an empty list. A list of response headers to expose to CORS clients. Limited to 64 defined headers and two prefixed headers. Each header can be up to 256 characters.
- **allowed_headers** (*list of str*) – Defaults to an empty list. A list of headers allowed to be part of the cross-origin request. Limited to 64 defined headers and 2 prefixed headers. Each header can be up to 256 characters.

```
class azure.storage.models.GeoReplication
```

Bases: object

Contains statistics related to replication for the given service.

Variables

- **status** (*str*) –

The status of the secondary location. Possible values are: live: Indicates that the secondary location is active and operational. bootstrap: Indicates initial synchronization from the primary location

to the secondary location is in progress. This typically occurs when replication is first enabled.

unavailable: Indicates that the secondary location is temporarily unavailable.

- **last_sync_time** (*date*) – A GMT date value, to the second. All primary writes preceding this value are guaranteed to be available for read operations at the secondary. Primary writes after this point in time may or may not be available for reads. The value may be empty if LastSyncTime is not available. This can happen if the replication status is bootstrap or unavailable. Although geo-replication is continuously enabled, the LastSyncTime result may reflect a cached value from the service that is refreshed every few minutes.

```
class azure.storage.models.ListGenerator(resources, list_method, list_args, list_kwargs)
```

Bases: `abcollections.Iterable`

A generator object used to list storage resources. The generator will lazily follow the continuation tokens returned by the service and stop when all resources have been returned or `max_results` is reached.

If `max_results` is specified and the account has more than that number of resources, the generator will have a populated `next_marker` field once it finishes. This marker can be used to create a new generator if more results are desired.

class `azure.storage.models.LocationMode`

Bases: `object`

Specifies the location the request should be sent to. This mode only applies for RA-GRS accounts which allow secondary read access. All other account types must use PRIMARY.

PRIMARY = 'primary'

Requests should be sent to the primary location.

SECONDARY = 'secondary'

Requests should be sent to the secondary location, if possible.

class `azure.storage.models.Logging` (*delete=False, read=False, write=False, retention_policy=None*)

Bases: `object`

Storage Analytics logs detailed information about successful and failed requests to a storage service. This information can be used to monitor individual requests and to diagnose issues with a storage service. Requests are logged on a best-effort basis.

All logs are stored in block blobs in a container named \$logs, which is automatically created when Storage Analytics is enabled for a storage account. The \$logs container is located in the blob namespace of the storage account. This container cannot be deleted once Storage Analytics has been enabled, though its contents can be deleted.

For more information, see <https://msdn.microsoft.com/en-us/library/azure/hh343262.aspx>

Parameters

- **delete** (*bool*) – Indicates whether all delete requests should be logged.
- **read** (*bool*) – Indicates whether all read requests should be logged.
- **write** (*bool*) – Indicates whether all write requests should be logged.
- **retention_policy** (*RetentionPolicy*) – The retention policy for the metrics.

class `azure.storage.models.Metrics` (*enabled=False, include_apis=None, retention_policy=None*)

Bases: `object`

Metrics include aggregated transaction statistics and capacity data about requests to a storage service. Transactions are reported at both the API operation level as well as at the storage service level, and capacity is reported at the storage service level. Metrics data can be used to analyze storage service usage, diagnose issues with requests made against the storage service, and to improve the performance of applications that use a service.

For more information, see <https://msdn.microsoft.com/en-us/library/azure/hh343258.aspx>

Parameters

- **enabled** (*bool*) – Indicates whether metrics are enabled for the service.
- **include_apis** (*bool*) – Required if enabled is True. Indicates whether metrics should generate summary statistics for called API operations.
- **retention_policy** (*RetentionPolicy*) – The retention policy for the metrics.

class `azure.storage.models.Protocol`

Bases: `object`

Specifies the protocol permitted for a SAS token. Note that HTTP only is not allowed.

HTTPS = 'https'

Allow HTTPS requests only.

HTTPS_HTTP = 'https,http'

Allow HTTP and HTTPS requests.

class `azure.storage.models.ResourceTypes` (*service=False, container=False, object=False, _str=None*)

Bases: `object`

Specifies the resource types that are accessible with the account SAS.

Variables

- **`ResourceTypes.CONTAINER`** (`ResourceTypes`) – Access to container-level APIs (e.g., Create/Delete Container, Create/Delete Queue, Create/Delete Table, Create/Delete Share, List Blobs/Files and Directories)
- **`ResourceTypes.OBJECT`** (`ResourceTypes`) – Access to object-level APIs for blobs, queue messages, table entities, and files(e.g. Put Blob, Query Entity, Get Messages, Create File, etc.)
- **`ResourceTypes.SERVICE`** (`ResourceTypes`) – Access to service-level APIs (e.g., Get/Set Service Properties, Get Service Stats, List Containers/Queues/Tables/Shares)

Parameters

- **`service`** (*bool*) – Access to service-level APIs (e.g., Get/Set Service Properties, Get Service Stats, List Containers/Queues/Tables/Shares)
- **`container`** (*bool*) – Access to container-level APIs (e.g., Create/Delete Container, Create/Delete Queue, Create/Delete Table, Create/Delete Share, List Blobs/Files and Directories)
- **`object`** (*bool*) – Access to object-level APIs for blobs, queue messages, table entities, and files(e.g. Put Blob, Query Entity, Get Messages, Create File, etc.)
- **`_str`** (*str*) – A string representing the resource types.

CONTAINER = `<azure.storage.models.ResourceTypes object>`

OBJECT = `<azure.storage.models.ResourceTypes object>`

SERVICE = `<azure.storage.models.ResourceTypes object>`

class `azure.storage.models.RetentionPolicy` (*enabled=False, days=None*)

Bases: `object`

By default, Storage Analytics will not delete any logging or metrics data. Blobs and table entities will continue to be written until the shared 20TB limit is reached. Once the 20TB limit is reached, Storage Analytics will stop writing new data and will not resume until free space is available. This 20TB limit is independent of the total limit for your storage account.

There are two ways to delete Storage Analytics data: by manually making deletion requests or by setting a data retention policy. Manual requests to delete Storage Analytics data are billable, but delete requests resulting from a retention policy are not billable.

Parameters

- **`enabled`** (*bool*) – Indicates whether a retention policy is enabled for the storage service. If disabled, logging and metrics data will be retained infinitely by the service unless explicitly deleted.

- **days** (*int*) – Required if enabled is true. Indicates the number of days that metrics or logging data should be retained. All data older than this value will be deleted. The minimum value you can specify is 1; the largest value is 365 (one year).

class `azure.storage.models.RetryContext`

Bases: `object`

Contains the request and response information that can be used to determine whether and how to retry. This context is stored across retries and may be used to store other information relevant to the retry strategy.

:ivar HTTPRequest request: The request sent to the storage service.

:ivar HTTPResponse response: The response returned by the storage service.

Variables location_mode (`LocationMode`) – The location the request was sent to.

class `azure.storage.models.ServiceProperties`

Bases: `object`

Returned by `get_*_service_properties` functions. Contains the properties of a storage service, including Analytics and CORS rules.

Azure Storage Analytics performs logging and provides metrics data for a storage account. You can use this data to trace requests, analyze usage trends, and diagnose issues with your storage account. To use Storage Analytics, you must enable it individually for each service you want to monitor.

The aggregated data is stored in a well-known blob (for logging) and in well-known tables (for metrics), which may be accessed using the Blob service and Table service APIs.

For an in-depth guide on using Storage Analytics and other tools to identify, diagnose, and troubleshoot Azure Storage-related issues, see <http://azure.microsoft.com/documentation/articles/storage-monitoring-diagnosing-troubleshooting/>

For more information on CORS, see <https://msdn.microsoft.com/en-us/library/azure/dn535601.aspx>

class `azure.storage.models.ServiceStats`

Bases: `object`

Returned by `get_*_service_stats` functions. Contains statistics related to replication for the given service. It is only available when read-access geo-redundant replication is enabled for the storage account.

Variables geo_replication (`GeoReplication`) – An object containing statistics related to replication for the given service.

class `azure.storage.models.Services` (*blob=False, queue=False, table=False, file=False, _str=None*)

Bases: `object`

Specifies the services accessible with the account SAS.

Variables

- **Services.BLOB** (`Services`) – The blob service.
- **Services.FILE** (`Services`) – The file service
- **Services.QUEUE** (`Services`) – The queue service.
- **Services.TABLE** (`Services`) – The table service

Parameters

- **blob** (*bool*) – Access to any blob service, for example, the `.BlockBlobService`
- **queue** (*bool*) – Access to the `.QueueService`

- **table** (*bool*) – Access to the *.TableService*
- **file** (*bool*) – Access to the *.FileService*
- **_str** (*str*) – A string representing the services.

BLOB = <azure.storage.models.Services object>

FILE = <azure.storage.models.Services object>

QUEUE = <azure.storage.models.Services object>

TABLE = <azure.storage.models.Services object>

azure.storage.sharedaccesssignature module

class azure.storage.sharedaccesssignature.**SharedAccessSignature** (*account_name*,
account_key)

Bases: object

Provides a factory for creating blob, queue, table, and file shares access signature tokens with a common account name and account key. Users can either use the factory or can construct the appropriate service and use the generate_*_shared_access_signature method directly.

Parameters

- **account_name** (*str*) – The storage account name used to generate the shared access signatures.
- **account_key** (*str*) – The access key to generate the shares access signatures.

generate_account (*services*, *resource_types*, *permission*, *expiry*, *start=None*, *ip=None*, *protocol=None*)

Generates a shared access signature for the account. Use the returned signature with the sas_token parameter of the service or to create a new account object.

Parameters

- **services** (*Services*) – Specifies the services accessible with the account SAS. You can combine values to provide access to more than one service.
- **resource_types** (*ResourceTypes*) – Specifies the resource types that are accessible with the account SAS. You can combine values to provide access to more than one resource type.
- **permission** (*AccountPermissions*) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. You can combine values to provide more than one permission.
- **expiry** (*datetime.datetime or str*) – The time at which the shared access signature becomes invalid. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **start** (*datetime.datetime or str*) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.

- **ip** (*str*) – Specifies an IP address or a range of IP addresses from which to accept requests. If the IP address from which the request originates does not match the IP address or address range specified on the SAS token, the request is not authenticated. For example, specifying `sip=168.1.5.65` or `sip=168.1.5.60-168.1.5.70` on the SAS restricts the request to those IP addresses.
- **protocol** (*str*) – Specifies the protocol permitted for a request made. The default value is `https,http`. See *Protocol* for possible values.

generate_blob (*container_name*, *blob_name*, *permission=None*, *expiry=None*, *start=None*, *id=None*, *ip=None*, *protocol=None*, *cache_control=None*, *content_disposition=None*, *content_encoding=None*, *content_language=None*, *content_type=None*)

Generates a shared access signature for the blob. Use the returned signature with the `sas_token` parameter of any `BlobService`.

Parameters

- **container_name** (*str*) – Name of container.
- **blob_name** (*str*) – Name of blob.
- **permission** (`BlobPermissions`) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Permissions must be ordered read, write, delete, list. Required unless an `id` is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy.
- **expiry** (*datetime.datetime* or *str*) – The time at which the shared access signature becomes invalid. Required unless an `id` is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **start** (*datetime.datetime* or *str*) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **id** (*str*) – A unique value up to 64 characters in length that correlates to a stored access policy. To create a stored access policy, use `set_blob_service_properties`.
- **ip** (*str*) – Specifies an IP address or a range of IP addresses from which to accept requests. If the IP address from which the request originates does not match the IP address or address range specified on the SAS token, the request is not authenticated. For example, specifying `sip=168.1.5.65` or `sip=168.1.5.60-168.1.5.70` on the SAS restricts the request to those IP addresses.
- **protocol** (*str*) – Specifies the protocol permitted for a request made. The default value is `https,http`. See *Protocol* for possible values.
- **cache_control** (*str*) – Response header value for Cache-Control when resource is accessed using this shared access signature.
- **content_disposition** (*str*) – Response header value for Content-Disposition when resource is accessed using this shared access signature.
- **content_encoding** (*str*) – Response header value for Content-Encoding when resource is accessed using this shared access signature.
- **content_language** (*str*) – Response header value for Content-Language when resource is accessed using this shared access signature.

- **content_type** (*str*) – Response header value for Content-Type when resource is accessed using this shared access signature.

generate_container (*container_name, permission=None, expiry=None, start=None, id=None, ip=None, protocol=None, cache_control=None, content_disposition=None, content_encoding=None, content_language=None, content_type=None*)

Generates a shared access signature for the container. Use the returned signature with the `sas_token` parameter of any `BlobService`.

Parameters

- **container_name** (*str*) – Name of container.
- **permission** (`ContainerPermissions`) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Permissions must be ordered read, write, delete, list. Required unless an `id` is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy.
- **expiry** (`datetime.datetime` or *str*) – The time at which the shared access signature becomes invalid. Required unless an `id` is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **start** (`datetime.datetime` or *str*) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **id** (*str*) – A unique value up to 64 characters in length that correlates to a stored access policy. To create a stored access policy, use `set_blob_service_properties`.
- **ip** (*str*) – Specifies an IP address or a range of IP addresses from which to accept requests. If the IP address from which the request originates does not match the IP address or address range specified on the SAS token, the request is not authenticated. For example, specifying `sip=168.1.5.65` or `sip=168.1.5.60-168.1.5.70` on the SAS restricts the request to those IP addresses.
- **protocol** (*str*) – Specifies the protocol permitted for a request made. The default value is `https,http`. See [Protocol](#) for possible values.
- **cache_control** (*str*) – Response header value for Cache-Control when resource is accessed using this shared access signature.
- **content_disposition** (*str*) – Response header value for Content-Disposition when resource is accessed using this shared access signature.
- **content_encoding** (*str*) – Response header value for Content-Encoding when resource is accessed using this shared access signature.
- **content_language** (*str*) – Response header value for Content-Language when resource is accessed using this shared access signature.
- **content_type** (*str*) – Response header value for Content-Type when resource is accessed using this shared access signature.

generate_file (*share_name, directory_name=None, file_name=None, permission=None, expiry=None, start=None, id=None, ip=None, protocol=None, cache_control=None, content_disposition=None, content_encoding=None, content_language=None, content_type=None*)

Generates a shared access signature for the file. Use the returned signature with the `sas_token` parameter

of FileService.

Parameters

- **share_name** (*str*) – Name of share.
- **directory_name** (*str*) – Name of directory. SAS tokens cannot be created for directories, so this parameter should only be present if file_name is provided.
- **file_name** (*str*) – Name of file.
- **permission** (`FilePermissions`) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Permissions must be ordered read, create, write, delete, list. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy.
- **expiry** (*datetime.datetime* or *str*) – The time at which the shared access signature becomes invalid. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **start** (*datetime.datetime* or *str*) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **id** (*str*) – A unique value up to 64 characters in length that correlates to a stored access policy. To create a stored access policy, use `set_file_service_properties`.
- **ip** (*str*) – Specifies an IP address or a range of IP addresses from which to accept requests. If the IP address from which the request originates does not match the IP address or address range specified on the SAS token, the request is not authenticated. For example, specifying `sip=168.1.5.65` or `sip=168.1.5.60-168.1.5.70` on the SAS restricts the request to those IP addresses.
- **protocol** (*str*) – Specifies the protocol permitted for a request made. The default value is `https,http`. See *Protocol* for possible values.
- **cache_control** (*str*) – Response header value for Cache-Control when resource is accessed using this shared access signature.
- **content_disposition** (*str*) – Response header value for Content-Disposition when resource is accessed using this shared access signature.
- **content_encoding** (*str*) – Response header value for Content-Encoding when resource is accessed using this shared access signature.
- **content_language** (*str*) – Response header value for Content-Language when resource is accessed using this shared access signature.
- **content_type** (*str*) – Response header value for Content-Type when resource is accessed using this shared access signature.

generate_queue (*queue_name*, *permission=None*, *expiry=None*, *start=None*, *id=None*, *ip=None*, *protocol=None*)

Generates a shared access signature for the queue. Use the returned signature with the `sas_token` parameter of `QueueService`.

Parameters

- **queue_name** (*str*) – Name of queue.

- **permission** (`QueuePermissions`) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Permissions must be ordered read, add, update, process. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy.
- **expiry** (`datetime.datetime` or `str`) – The time at which the shared access signature becomes invalid. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **start** (`datetime.datetime` or `str`) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **id** (`str`) – A unique value up to 64 characters in length that correlates to a stored access policy. To create a stored access policy, use `set_blob_service_properties`.
- **ip** (`str`) – Specifies an IP address or a range of IP addresses from which to accept requests. If the IP address from which the request originates does not match the IP address or address range specified on the SAS token, the request is not authenticated. For example, specifying `sip=168.1.5.65` or `sip=168.1.5.60-168.1.5.70` on the SAS restricts the request to those IP addresses.
- **protocol** (`str`) – Specifies the protocol permitted for a request made. The default value is `https,http`. See [Protocol](#) for possible values.

generate_share (`share_name`, `permission=None`, `expiry=None`, `start=None`, `id=None`, `ip=None`, `protocol=None`, `cache_control=None`, `content_disposition=None`, `content_encoding=None`, `content_language=None`, `content_type=None`)

Generates a shared access signature for the share. Use the returned signature with the `sas_token` parameter of `FileService`.

Parameters

- **share_name** (`str`) – Name of share.
- **permission** (`SharePermissions`) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Permissions must be ordered read, create, write, delete, list. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy.
- **expiry** (`datetime.datetime` or `str`) – The time at which the shared access signature becomes invalid. Required unless an id is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **start** (`datetime.datetime` or `str`) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **id** (`str`) – A unique value up to 64 characters in length that correlates to a stored access policy. To create a stored access policy, use `set_file_service_properties`.
- **ip** (`str`) – Specifies an IP address or a range of IP addresses from which to accept requests. If the IP address from which the request originates does not match the IP address

or address range specified on the SAS token, the request is not authenticated. For example, specifying `sip=168.1.5.65` or `sip=168.1.5.60-168.1.5.70` on the SAS restricts the request to those IP addresses.

- **protocol** (*str*) – Specifies the protocol permitted for a request made. The default value is `https,http`. See *Protocol* for possible values.
- **cache_control** (*str*) – Response header value for Cache-Control when resource is accessed using this shared access signature.
- **content_disposition** (*str*) – Response header value for Content-Disposition when resource is accessed using this shared access signature.
- **content_encoding** (*str*) – Response header value for Content-Encoding when resource is accessed using this shared access signature.
- **content_language** (*str*) – Response header value for Content-Language when resource is accessed using this shared access signature.
- **content_type** (*str*) – Response header value for Content-Type when resource is accessed using this shared access signature.

generate_table (*table_name, permission=None, expiry=None, start=None, id=None, ip=None, protocol=None, start_pk=None, start_rk=None, end_pk=None, end_rk=None*)

Generates a shared access signature for the table. Use the returned signature with the `sas_token` parameter of `TableService`.

Parameters

- **table_name** (*str*) – Name of table.
- **permission** (`TablePermissions`) – The permissions associated with the shared access signature. The user is restricted to operations allowed by the permissions. Required unless an `id` is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy.
- **expiry** (`datetime.datetime` or *str*) – The time at which the shared access signature becomes invalid. Required unless an `id` is given referencing a stored access policy which contains this field. This field must be omitted if it has been specified in an associated stored access policy. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **start** (`datetime.datetime` or *str*) – The time at which the shared access signature becomes valid. If omitted, start time for this call is assumed to be the time when the storage service receives the request. Azure will always convert values to UTC. If a date is passed in without timezone info, it is assumed to be UTC.
- **id** (*str*) – A unique value up to 64 characters in length that correlates to a stored access policy. To create a stored access policy, use `set_blob_service_properties`.
- **ip** (*str*) – Specifies an IP address or a range of IP addresses from which to accept requests. If the IP address from which the request originates does not match the IP address or address range specified on the SAS token, the request is not authenticated. For example, specifying `sip=168.1.5.65` or `sip=168.1.5.60-168.1.5.70` on the SAS restricts the request to those IP addresses.
- **protocol** (*str*) – Specifies the protocol permitted for a request made. The default value is `https,http`. See *Protocol* for possible values.
- **start_pk** (*str*) – The minimum partition key accessible with this shared access signature. `startpk` must accompany `startrk`. Key values are inclusive. If omitted, there is no lower bound on the table entities that can be accessed.

- **start_rk** (*str*) – The minimum row key accessible with this shared access signature. startpk must accompany startrk. Key values are inclusive. If omitted, there is no lower bound on the table entities that can be accessed.
- **end_pk** (*str*) – The maximum partition key accessible with this shared access signature. endpk must accompany endrk. Key values are inclusive. If omitted, there is no upper bound on the table entities that can be accessed.
- **end_rk** (*str*) – The maximum row key accessible with this shared access signature. endpk must accompany endrk. Key values are inclusive. If omitted, there is no upper bound on the table entities that can be accessed.

azure.storage.storageclient module

class `azure.storage.storageclient.StorageClient` (*connection_params*)

Bases: `object`

This is the base class for service objects. Service objects are used to do all requests to Storage. This class cannot be instantiated directly.

Variables

- **account_name** (*str*) – The storage account name. This is used to authenticate requests signed with an account key and to construct the storage endpoint. It is required unless a connection string is given, or if a custom domain is used with anonymous authentication.
- **account_key** (*str*) – The storage account key. This is used for shared key authentication. If neither account key or sas token is specified, anonymous access will be used.
- **sas_token** (*str*) – A shared access signature token to use to authenticate requests instead of the account key. If account key and sas token are both specified, account key will be used to sign. If neither are specified, anonymous access will be used.
- **primary_endpoint** (*str*) – The endpoint to send storage requests to.
- **secondary_endpoint** (*str*) – The secondary endpoint to read storage data from. This will only be a valid endpoint if the storage account used is RA-GRS and thus allows reading from secondary.
- **retry** (*function(context)*) – A function which determines whether to retry. Takes as a parameter a `RetryContext` object. Returns the number of seconds to wait before retrying the request, or None to indicate not to retry.
- **location_mode** (`LocationMode`) – The host location to use to make requests. Defaults to `LocationMode.PRIMARY`. Note that this setting only applies to RA-GRS accounts as other account types do not allow reading from secondary. If the `location_mode` is set to `LocationMode.SECONDARY`, read requests will be sent to the secondary endpoint. Write requests will continue to be sent to primary.
- **protocol** (*str*) – The protocol to use for requests. Defaults to https.
- **request_session** (`requests.Session`) – The session object to use for http requests.
- **request_callback** (*function(request)*) – A function called immediately before each request is sent. This function takes as a parameter the request object and returns nothing. It may be used to added custom headers or log request data.

- **response_callback** (*function()*) – A function called immediately after each response is received. This function takes as a parameter the response object and returns nothing. It may be used to log response data.
- **retry_callback** (*function()*) – A function called immediately after retry evaluation is performed. This function takes as a parameter the retry context object and returns nothing. It may be used to detect retries and log context information.

Parameters **connection_params** (*obj*) – The parameters to use to construct the client.

protocol

request_session

set_proxy (*host, port, user=None, password=None*)

Sets the proxy server host and port for the HTTP CONNECT Tunnelling.

Parameters

- **host** (*str*) – Address of the proxy. Ex: '192.168.0.100'
- **port** (*int*) – Port of the proxy. Ex: 6000
- **user** (*str*) – User for proxy authorization.
- **password** (*str*) – Password for proxy authorization.

socket_timeout

Module contents

Module contents

azure

a

azure, 142
azure.storage, 142
azure.storage.blob, 79
azure.storage.blob.appendblob, 17
azure.storage.blob.baseblob, 22
azure.storage.blob.blockblob, 49
azure.storage.blob.models, 57
azure.storage.blob.pageblob, 65
azure.storage.cloudstorageaccount, 127
azure.storage.file, 103
azure.storage.file.fileservice, 79
azure.storage.file.models, 98
azure.storage.models, 129
azure.storage.queue, 114
azure.storage.queue.models, 103
azure.storage.queue.queue, 105
azure.storage.sharedaccesssignature, 135
azure.storage.storageclient, 141
azure.storage.table, 127
azure.storage.table.models, 114
azure.storage.table.tablebatch, 116
azure.storage.table.tableservice, 118

A

- abort_copy_blob() (azure.storage.blob.baseblobservice.BaseBlobService method), 24
- abort_copy_file() (azure.storage.file.fileservice.FileService method), 80
- AccessPolicy (class in azure.storage.models), 129
- AccountPermissions (class in azure.storage.models), 129
- acquire_blob_lease() (azure.storage.blob.baseblobservice.BaseBlobService method), 24
- acquire_container_lease() (azure.storage.blob.baseblobservice.BaseBlobService method), 25
- ADD (azure.storage.blob.models.BlobPermissions attribute), 59
- ADD (azure.storage.models.AccountPermissions attribute), 130
- ADD (azure.storage.queue.models.QueuePermissions attribute), 105
- ADD (azure.storage.table.models.TablePermissions attribute), 116
- All (azure.storage.blob.models.BlockListType attribute), 60
- append_blob_from_bytes() (azure.storage.blob.appendblobservice.AppendBlobService method), 18
- append_blob_from_path() (azure.storage.blob.appendblobservice.AppendBlobService method), 18
- append_blob_from_stream() (azure.storage.blob.appendblobservice.AppendBlobService method), 19
- append_blob_from_text() (azure.storage.blob.appendblobservice.AppendBlobService method), 20
- append_block() (azure.storage.blob.appendblobservice.AppendBlobService method), 20
- AppendBlobService (class in azure.storage.blob.appendblobservice), 17
- AppendBlockProperties (class in azure.storage.blob.models), 57
- Archive (azure.storage.blob.models.StandardBlobTier attribute), 65
- azure (module), 142
- azure.storage (module), 142
- azure.storage.blob (module), 79
- azure.storage.blob.appendblobservice (module), 17
- azure.storage.blob.baseblobservice (module), 22
- azure.storage.blob.blockblobservice (module), 49
- azure.storage.blob.models (module), 57
- azure.storage.blob.pageblobservice (module), 65
- azure.storage.cloudstorageaccount (module), 127
- azure.storage.file (module), 103
- azure.storage.file.fileservice (module), 79
- azure.storage.file.models (module), 98
- azure.storage.models (module), 129
- azure.storage.queue (module), 114
- azure.storage.queue.models (module), 103
- azure.storage.queue.queue.service (module), 105
- azure.storage.sharedaccesssignature (module), 135
- azure.storage.storageclient (module), 141
- azure.storage.table (module), 127
- azure.storage.table.models (module), 114
- azure.storage.table.tablebatch (module), 116
- azure.storage.table.tableservice (module), 118
- AzureBatchOperationError, 114
- AzureBatchValidationError, 114

B

- BaseBlobService (class in azure.storage.blob.baseblobservice), 22
- batch() (azure.storage.table.tableservice.TableService method), 119
- BINARY (azure.storage.table.models.EdmType attribute), 114
- binary_base64decode() (azure.storage.queue.models.QueueMessageFormat static method), 104
- binary_base64encode() (azure.storage.queue.models.QueueMessageFormat static method), 104

- Blob (azure.storage.blob.models.PublicAccess attribute), 64
- BLOB (azure.storage.models.Services attribute), 135
- Blob (class in azure.storage.blob.models), 57
- BlobBlock (class in azure.storage.blob.models), 58
- BlobBlockList (class in azure.storage.blob.models), 58
- BlobBlockState (class in azure.storage.blob.models), 58
- BlobPermissions (class in azure.storage.blob.models), 58
- BlobPrefix (class in azure.storage.blob.models), 59
- BlobProperties (class in azure.storage.blob.models), 59
- BlockBlobService (class in azure.storage.blob.blockblobservice), 49
- BlockListType (class in azure.storage.blob.models), 60
- BOOLEAN (azure.storage.table.models.EdmType attribute), 114
- break_blob_lease() (azure.storage.blob.baseblobservice.BaseBlobService method), 25
- break_container_lease() (azure.storage.blob.baseblobservice.BaseBlobService method), 26
- C**
- change_blob_lease() (azure.storage.blob.baseblobservice.BaseBlobService method), 27
- change_container_lease() (azure.storage.blob.baseblobservice.BaseBlobService method), 27
- clear_messages() (azure.storage.queue.queue.service.QueueService method), 107
- clear_page() (azure.storage.blob.pageblobservice.PageBlobService method), 66
- clear_range() (azure.storage.file.fileservice.FileService method), 81
- CloudStorageAccount (class in azure.storage.cloudstorageaccount), 127
- commit_batch() (azure.storage.table.tableservice.TableService method), 119
- Committed (azure.storage.blob.models.BlobBlockState attribute), 58
- Committed (azure.storage.blob.models.BlockListType attribute), 60
- Container (azure.storage.blob.models.PublicAccess attribute), 64
- CONTAINER (azure.storage.models.ResourceTypes attribute), 133
- Container (class in azure.storage.blob.models), 60
- ContainerPermissions (class in azure.storage.blob.models), 60
- ContainerProperties (class in azure.storage.blob.models), 61
- ContentSettings (class in azure.storage.blob.models), 61
- ContentSettings (class in azure.storage.file.models), 98
- Cool (azure.storage.blob.models.StandardBlobTier attribute), 65
- COPY (azure.storage.blob.models.Include attribute), 63
- copy_blob() (azure.storage.blob.baseblobservice.BaseBlobService method), 28
- copy_blob() (azure.storage.blob.pageblobservice.PageBlobService method), 67
- copy_file() (azure.storage.file.fileservice.FileService method), 81
- CopyProperties (class in azure.storage.blob.models), 62
- CopyProperties (class in azure.storage.file.models), 99
- CorsRule (class in azure.storage.models), 131
- CREATE (azure.storage.blob.models.BlobPermissions attribute), 59
- CREATE (azure.storage.file.models.FilePermissions attribute), 101
- CREATE (azure.storage.models.AccountPermissions attribute), 130
- create_blob_service() (azure.storage.cloudstorageaccount.CloudStorageAccount method), 127
- create_blob() (azure.storage.blob.appendblobservice.AppendBlobService method), 21
- create_blob() (azure.storage.blob.pageblobservice.PageBlobService method), 69
- create_blob_from_bytes() (azure.storage.blob.blockblobservice.BlockBlobService method), 50
- create_blob_from_bytes() (azure.storage.blob.pageblobservice.PageBlobService method), 70
- create_blob_from_path() (azure.storage.blob.blockblobservice.BlockBlobService method), 51
- create_blob_from_path() (azure.storage.blob.pageblobservice.PageBlobService method), 71
- create_blob_from_stream() (azure.storage.blob.blockblobservice.BlockBlobService method), 52
- create_blob_from_stream() (azure.storage.blob.pageblobservice.PageBlobService method), 72
- create_blob_from_text() (azure.storage.blob.blockblobservice.BlockBlobService method), 54
- create_block_blob_service() (azure.storage.cloudstorageaccount.CloudStorageAccount method), 128
- create_container() (azure.storage.blob.baseblobservice.BaseBlobService method), 30
- create_directory() (azure.storage.file.fileservice.FileService method), 81
- create_file() (azure.storage.file.fileservice.FileService method), 82
- create_file_from_bytes() (azure.storage.file.fileservice.FileService method), 82
- create_file_from_path() (azure.storage.file.fileservice.FileService method), 82

method), 83
 create_file_from_stream()
 (azure.storage.file.fileservice.FileService
 method), 83
 create_file_from_text() (azure.storage.file.fileservice.FileService
 method), 84
 create_file_service() (azure.storage.cloudstorageaccount.CloudStorageAccount
 method), 128
 create_page_blob_service()
 (azure.storage.cloudstorageaccount.CloudStorageAccount
 method), 128
 create_queue() (azure.storage.queue.queue.service.QueueService
 method), 107
 create_queue_service() (azure.storage.cloudstorageaccount.CloudStorageAccount
 method), 128
 create_share() (azure.storage.file.fileservice.FileService
 method), 85
 create_table() (azure.storage.table.tableservice.TableService
 method), 119
 create_table_service() (azure.storage.cloudstorageaccount.CloudStorageAccount
 method), 128

D

DATETIME (azure.storage.table.models.EdmType
 attribute), 114
 DELETE (azure.storage.blob.models.BlobPermissions
 attribute), 59
 DELETE (azure.storage.blob.models.ContainerPermissions
 attribute), 61
 DELETE (azure.storage.file.models.FilePermissions
 attribute), 101
 DELETE (azure.storage.file.models.SharePermissions
 attribute), 102
 DELETE (azure.storage.models.AccountPermissions
 attribute), 130
 DELETE (azure.storage.table.models.TablePermissions
 attribute), 116
 delete_blob() (azure.storage.blob.baseblob.service.BaseBlobService
 method), 30
 delete_container() (azure.storage.blob.baseblob.service.BaseBlobService
 method), 31
 delete_directory() (azure.storage.file.fileservice.FileService
 method), 85
 delete_entity() (azure.storage.table.tablebatch.TableBatch
 method), 116
 delete_entity() (azure.storage.table.tableservice.TableService
 method), 120
 delete_file() (azure.storage.file.fileservice.FileService
 method), 85
 delete_message() (azure.storage.queue.queue.service.QueueService
 method), 107
 delete_queue() (azure.storage.queue.queue.service.QueueService
 method), 108

delete_share() (azure.storage.file.fileservice.FileService
 method), 85
 delete_table() (azure.storage.table.tableservice.TableService
 method), 120
 DeleteSnapshot (class in azure.storage.blob.models), 62
 Directory (class in azure.storage.file.models), 99
 DiskStorageAccount (class in azure.storage.file.models),
 100
 DOUBLE (azure.storage.table.models.EdmType
 attribute), 115

E

EdmType (class in azure.storage.table.models), 114
 CloudStorageAccount (class in azure.storage.table.models), 115
 EntityProperty (class in azure.storage.table.models), 115
 exists() (azure.storage.blob.baseblob.service.BaseBlobService
 method), 31
 exists() (azure.storage.file.fileservice.FileService
 method), 86
 exists() (azure.storage.queue.queue.service.QueueService
 method), 108
 exists() (azure.storage.table.tableservice.TableService
 method), 120

F

FILE (azure.storage.models.Services
 attribute), 135
 File (class in azure.storage.file.models), 100
 FilePermissions (class in azure.storage.file.models), 100
 FileProperties (class in azure.storage.file.models), 101
 FileRange (class in azure.storage.file.models), 101
 FileService (class in azure.storage.file.fileservice), 79

G

generate_account() (azure.storage.sharedaccesssignature.SharedAccessSignature
 method), 135
 generate_account_shared_access_signature()
 (azure.storage.blob.baseblob.service.BaseBlobService
 method), 31
 generate_account_shared_access_signature()
 (azure.storage.file.fileservice.FileService
 method), 86
 generate_account_shared_access_signature()
 (azure.storage.queue.queue.service.QueueService
 method), 108
 generate_account_shared_access_signature()
 (azure.storage.table.tableservice.TableService
 method), 121
 generate_blob() (azure.storage.sharedaccesssignature.SharedAccessSignature
 method), 136
 generate_blob_shared_access_signature()
 (azure.storage.blob.baseblob.service.BaseBlobService
 method), 32
 generate_container() (azure.storage.sharedaccesssignature.SharedAccessSignature
 method), 137

generate_container_shared_access_signature() (azure.storage.blob.baseblobservice.BaseBlobService method), 33
 generate_file() (azure.storage.sharedaccesssignature.SharedAccessSignature method), 137
 generate_file_shared_access_signature() (azure.storage.file.fileservice.FileService method), 87
 generate_queue() (azure.storage.sharedaccesssignature.SharedAccessSignature method), 138
 generate_queue_shared_access_signature() (azure.storage.queue.queue.service.QueueService method), 109
 generate_share() (azure.storage.sharedaccesssignature.SharedAccessSignature method), 139
 generate_share_shared_access_signature() (azure.storage.file.fileservice.FileService method), 88
 generate_shared_access_signature() (azure.storage.cloudstorageaccount.CloudStorageAccount method), 128
 generate_table() (azure.storage.sharedaccesssignature.SharedAccessSignature method), 140
 generate_table_shared_access_signature() (azure.storage.table.tableservice.TableService method), 121
 GeoReplication (class in azure.storage.models), 131
 get_blob_metadata() (azure.storage.blob.baseblobservice.BaseBlobService method), 34
 get_blob_properties() (azure.storage.blob.baseblobservice.BaseBlobService method), 35
 get_blob_service_properties() (azure.storage.blob.baseblobservice.BaseBlobService method), 35
 get_blob_service_stats() (azure.storage.blob.baseblobservice.BaseBlobService method), 36
 get_blob_to_bytes() (azure.storage.blob.baseblobservice.BaseBlobService method), 36
 get_blob_to_path() (azure.storage.blob.baseblobservice.BaseBlobService method), 37
 get_blob_to_stream() (azure.storage.blob.baseblobservice.BaseBlobService method), 39
 get_blob_to_text() (azure.storage.blob.baseblobservice.BaseBlobService method), 40
 get_block_list() (azure.storage.blob.blockblobservice.BlockBlobService method), 55
 get_container_acl() (azure.storage.blob.baseblobservice.BaseBlobService method), 41
 get_container_metadata() (azure.storage.blob.baseblobservice.BaseBlobService method), 42
 get_container_properties() (azure.storage.blob.baseblobservice.BaseBlobService method), 42
 get_directory_metadata() (azure.storage.file.fileservice.FileService method), 89
 get_entity() (azure.storage.table.tableservice.TableService method), 122
 get_file_access_sas() (azure.storage.file.fileservice.FileService method), 89
 get_file_properties() (azure.storage.file.fileservice.FileService method), 89
 get_file_service_properties() (azure.storage.file.fileservice.FileService method), 90
 get_file_to_bytes() (azure.storage.file.fileservice.FileService method), 90
 get_file_to_path() (azure.storage.file.fileservice.FileService method), 91
 get_file_to_stream() (azure.storage.file.fileservice.FileService method), 92
 get_files_signature() (azure.storage.file.fileservice.FileService method), 93
 get_messages() (azure.storage.queue.queue.service.QueueService method), 109
 get_page_ranges() (azure.storage.blob.pageblobservice.PageBlobService method), 73
 get_page_ranges_diff() (azure.storage.blob.pageblobservice.PageBlobService method), 74
 get_queue_metadata() (azure.storage.queue.queue.service.QueueService method), 110
 get_queue_service_properties() (azure.storage.queue.queue.service.QueueService method), 110
 get_queue_stats() (azure.storage.queue.queue.service.QueueService method), 110
 get_queue_service_stats() (azure.storage.queue.queue.service.QueueService method), 110
 get_share_acl() (azure.storage.file.fileservice.FileService method), 94
 get_share_metadata() (azure.storage.file.fileservice.FileService method), 94
 get_share_properties() (azure.storage.file.fileservice.FileService method), 94
 get_share_stats() (azure.storage.file.fileservice.FileService method), 94
 get_table_acl() (azure.storage.table.tableservice.TableService method), 123
 get_table_service_properties() (azure.storage.table.tableservice.TableService method), 123
 get_table_service_stats() (azure.storage.table.tableservice.TableService method), 123

- GUID (azure.storage.table.models.EdmType attribute), 115
- ## H
- Hot (azure.storage.blob.models.StandardBlobTier attribute), 65
- HTTPS (azure.storage.models.Protocol attribute), 133
- HTTPS_HTTP (azure.storage.models.Protocol attribute), 133
- ## I
- Include (azure.storage.blob.models.DeleteSnapshot attribute), 62
- Include (class in azure.storage.blob.models), 62
- Increment (azure.storage.blob.models.SequenceNumberAction attribute), 65
- incremental_copy_blob() (azure.storage.blob.pageblob.service.PageBlobService method), 75
- insert_entity() (azure.storage.table.tablebatch.TableBatch method), 117
- insert_entity() (azure.storage.table.tableservice.TableService method), 123
- insert_or_merge_entity() (azure.storage.table.tablebatch.TableBatch method), 117
- insert_or_merge_entity() (azure.storage.table.tableservice.TableService method), 124
- insert_or_replace_entity() (azure.storage.table.tablebatch.TableBatch method), 117
- insert_or_replace_entity() (azure.storage.table.tableservice.TableService method), 124
- INT32 (azure.storage.table.models.EdmType attribute), 115
- INT64 (azure.storage.table.models.EdmType attribute), 115
- ## J
- JSON_FULL_METADATA (azure.storage.table.models.TablePayloadFormat attribute), 115
- JSON_MINIMAL_METADATA (azure.storage.table.models.TablePayloadFormat attribute), 116
- JSON_NO_METADATA (azure.storage.table.models.TablePayloadFormat attribute), 116
- ## L
- Latest (azure.storage.blob.models.BlobBlockState attribute), 58
- LeaseProperties (class in azure.storage.blob.models), 63
- LIST (azure.storage.blob.models.ContainerPermissions attribute), 61
- LIST (azure.storage.file.models.SharePermissions attribute), 102
- LIST (azure.storage.models.AccountPermissions attribute), 130
- list_blobs() (azure.storage.blob.baseblob.service.BaseBlobService method), 42
- list_containers() (azure.storage.blob.baseblob.service.BaseBlobService method), 43
- list_directories_and_files() (azure.storage.file.fileservice.FileService method), 94
- list_queues() (azure.storage.queue.queue.service.QueueService method), 111
- list_ranges() (azure.storage.file.fileservice.FileService method), 95
- list_shares() (azure.storage.file.fileservice.FileService method), 95
- list_tables() (azure.storage.table.tableservice.TableService method), 124
- ListGenerator (class in azure.storage.models), 131
- LocationMode (class in azure.storage.models), 132
- Logging (class in azure.storage.models), 132
- ## M
- make_blob_url() (azure.storage.blob.baseblob.service.BaseBlobService method), 43
- make_file_url() (azure.storage.file.fileservice.FileService method), 96
- Max (azure.storage.blob.models.SequenceNumberAction attribute), 65
- MAX_BLOCK_SIZE (azure.storage.blob.appendblob.service.AppendBlobService attribute), 18
- MAX_BLOCK_SIZE (azure.storage.blob.blockblob.service.BlockBlobService attribute), 50
- MAX_CHUNK_GET_SIZE (azure.storage.blob.baseblob.service.BaseBlobService attribute), 24
- MAX_CHUNK_GET_SIZE (azure.storage.file.fileservice.FileService attribute), 80
- MAX_PAGE_SIZE (azure.storage.blob.pageblob.service.PageBlobService attribute), 66
- MAX_RANGE_SIZE (azure.storage.file.fileservice.FileService attribute), 80
- MAX_SINGLE_GET_SIZE (azure.storage.blob.baseblob.service.BaseBlobService attribute), 24
- MAX_SINGLE_GET_SIZE (azure.storage.file.fileservice.FileService attribute), 80

- MAX_SINGLE_PUT_SIZE (azure.storage.blob.blockblobservice.BlockBlobService attribute), 50
- merge_entity() (azure.storage.table.tablebatch.TableBatch method), 117
- merge_entity() (azure.storage.table.tableservice.TableService method), 125
- METADATA (azure.storage.blob.models.Include attribute), 63
- Metrics (class in azure.storage.models), 132
- MIN_LARGE_BLOCK_UPLOAD_THRESHOLD (azure.storage.blob.blockblobservice.BlockBlobService attribute), 50
- ## N
- nodecode() (azure.storage.queue.models.QueueMessageFormat static method), 104
- noencode() (azure.storage.queue.models.QueueMessageFormat static method), 104
- ## O
- OBJECT (azure.storage.models.ResourceTypes attribute), 133
- OFF (azure.storage.blob.models.PublicAccess attribute), 64
- Only (azure.storage.blob.models.DeleteSnapshot attribute), 62
- ## P
- P10 (azure.storage.blob.models.PremiumPageBlobTier attribute), 64
- P20 (azure.storage.blob.models.PremiumPageBlobTier attribute), 64
- P30 (azure.storage.blob.models.PremiumPageBlobTier attribute), 64
- P4 (azure.storage.blob.models.PremiumPageBlobTier attribute), 64
- P40 (azure.storage.blob.models.PremiumPageBlobTier attribute), 64
- P50 (azure.storage.blob.models.PremiumPageBlobTier attribute), 64
- P6 (azure.storage.blob.models.PremiumPageBlobTier attribute), 64
- P60 (azure.storage.blob.models.PremiumPageBlobTier attribute), 64
- PageBlobProperties (class in azure.storage.blob.models), 63
- PageBlobService (class in azure.storage.blob.pageblobservice), 65
- PageRange (class in azure.storage.blob.models), 63
- peek_messages() (azure.storage.queue.queue.service.QueueService method), 111
- PremiumPageBlobTier (class in azure.storage.blob.models), 64
- PRIMARY (azure.storage.models.LocationMode attribute), 132
- PROCESS (azure.storage.models.AccountPermissions attribute), 130
- PROCESS (azure.storage.queue.models.QueuePermissions attribute), 105
- protocol (azure.storage.storageclient.StorageClient attribute), 142
- Protocol (class in azure.storage.models), 132
- PublicAccess (class in azure.storage.blob.models), 64
- put_block() (azure.storage.blob.blockblobservice.BlockBlobService method), 55
- put_block_list() (azure.storage.blob.blockblobservice.BlockBlobService method), 56
- put_message() (azure.storage.queue.queue.service.QueueService method), 112
- ## Q
- QUERY (azure.storage.table.models.TablePermissions attribute), 116
- query_entities() (azure.storage.table.tableservice.TableService method), 125
- QUEUE (azure.storage.models.Services attribute), 135
- Queue (class in azure.storage.queue.models), 103
- QueueMessage (class in azure.storage.queue.models), 103
- QueueMessageFormat (class in azure.storage.queue.models), 103
- QueuePermissions (class in azure.storage.queue.models), 105
- QueueService (class in azure.storage.queue.queue.service), 105
- ## R
- READ (azure.storage.blob.models.BlobPermissions attribute), 59
- READ (azure.storage.blob.models.ContainerPermissions attribute), 61
- READ (azure.storage.file.models.FilePermissions attribute), 101
- READ (azure.storage.file.models.SharePermissions attribute), 102
- READ (azure.storage.models.AccountPermissions attribute), 130
- READ (azure.storage.queue.models.QueuePermissions attribute), 105
- release_blob_lease() (azure.storage.blob.baseblobservice.BaseBlobService method), 44
- release_container_lease() (azure.storage.blob.baseblobservice.BaseBlobService method), 44
- renew_blob_lease() (azure.storage.blob.baseblobservice.BaseBlobService method), 45

[renew_container_lease\(\)](#) (azure.storage.blob.baseblobservice.BaseBlobService method), 45
[request_session](#) (azure.storage.storageclient.StorageClient attribute), 142
[resize_blob\(\)](#) (azure.storage.blob.pageblobservice.PageBlobService method), 76
[resize_file\(\)](#) (azure.storage.file.fileservice.FileService method), 96
[ResourceProperties](#) (class in azure.storage.blob.models), 64
[ResourceTypes](#) (class in azure.storage.models), 133
[RetentionPolicy](#) (class in azure.storage.models), 133
[RetryContext](#) (class in azure.storage.models), 134
S
[SECONDARY](#) (azure.storage.models.LocationMode attribute), 132
[SequenceNumberAction](#) (class in azure.storage.blob.models), 65
[SERVICE](#) (azure.storage.models.ResourceTypes attribute), 133
[ServiceProperties](#) (class in azure.storage.models), 134
[Services](#) (class in azure.storage.models), 134
[ServiceStats](#) (class in azure.storage.models), 134
[set_blob_metadata\(\)](#) (azure.storage.blob.baseblobservice.BaseBlobService method), 46
[set_blob_properties\(\)](#) (azure.storage.blob.baseblobservice.BaseBlobService method), 46
[set_blob_service_properties\(\)](#) (azure.storage.blob.baseblobservice.BaseBlobService method), 47
[set_container_acl\(\)](#) (azure.storage.blob.baseblobservice.BaseBlobService method), 47
[set_container_metadata\(\)](#) (azure.storage.blob.baseblobservice.BaseBlobService method), 48
[set_directory_metadata\(\)](#) (azure.storage.file.fileservice.FileService method), 96
[set_file_metadata\(\)](#) (azure.storage.file.fileservice.FileService method), 96
[set_file_properties\(\)](#) (azure.storage.file.fileservice.FileService method), 97
[set_file_service_properties\(\)](#) (azure.storage.file.fileservice.FileService method), 97
[set_premium_page_blob_tier\(\)](#) (azure.storage.blob.pageblobservice.PageBlobService method), 77
[set_proxy\(\)](#) (azure.storage.storageclient.StorageClient method), 142
[set_queue_acl\(\)](#) (azure.storage.queue.queue.service.QueueService method), 112
[set_blob_service_metadata\(\)](#) (azure.storage.queue.queue.service.QueueService method), 113
[set_queue_service_properties\(\)](#) (azure.storage.queue.queue.service.QueueService method), 113
[set_sequence_number\(\)](#) (azure.storage.blob.pageblobservice.PageBlobService method), 77
[set_share_acl\(\)](#) (azure.storage.file.fileservice.FileService method), 97
[set_share_metadata\(\)](#) (azure.storage.file.fileservice.FileService method), 97
[set_share_properties\(\)](#) (azure.storage.file.fileservice.FileService method), 98
[set_standard_blob_tier\(\)](#) (azure.storage.blob.blockblobservice.BlockBlobService method), 57
[set_table_acl\(\)](#) (azure.storage.table.tableservice.TableService method), 126
[set_table_service_properties\(\)](#) (azure.storage.table.tableservice.TableService method), 126
[Share](#) (class in azure.storage.file.models), 101
[SharedAccessSignature](#) (class in azure.storage.sharedaccesssignature), 135
[SharePermissions](#) (class in azure.storage.file.models), 102
[ShareProperties](#) (class in azure.storage.file.models), 102
[snapshot_blob\(\)](#) (azure.storage.blob.baseblobservice.BaseBlobService method), 48
[SNAPSHOTS](#) (azure.storage.blob.models.Include attribute), 63
[socket_timeout](#) (azure.storage.storageclient.StorageClient attribute), 142
[StandardBlobTier](#) (class in azure.storage.blob.models), 65
[StorageClient](#) (class in azure.storage.storageclient), 141
[STRING](#) (azure.storage.table.models.EdmType attribute), 115
T
[TABLE](#) (azure.storage.models.Services attribute), 135
[Table](#) (class in azure.storage.table.models), 115
[TableBatch](#) (class in azure.storage.table.tablebatch), 116
[TablePayloadFormat](#) (class in azure.storage.table.models), 115
[TablePermissions](#) (class in azure.storage.table.models), 116
[TableService](#) (class in azure.storage.table.tableservice), 118
[text_base64decode\(\)](#) (azure.storage.queue.models.QueueMessageFormat static method), 104
[text_base64encode\(\)](#) (azure.storage.queue.models.QueueMessageFormat static method), 104
[text_xmldecode\(\)](#) (azure.storage.queue.models.QueueMessageFormat static method), 104

`text_xmlencode()` (`azure.storage.queue.models.QueueMessageFormat` static method), 104

U

`Uncommitted` (`azure.storage.blob.models.BlobBlockState` attribute), 58

`Uncommitted` (`azure.storage.blob.models.BlockListType` attribute), 60

`UNCOMMITTED_BLOBS` (`azure.storage.blob.models.Include` attribute), 63

`Update` (`azure.storage.blob.models.SequenceNumberAction` attribute), 65

`UPDATE` (`azure.storage.models.AccountPermissions` attribute), 130

`UPDATE` (`azure.storage.queue.models.QueuePermissions` attribute), 105

`UPDATE` (`azure.storage.table.models.TablePermissions` attribute), 116

`update_entity()` (`azure.storage.table.tablebatch.TableBatch` method), 117

`update_entity()` (`azure.storage.table.tableservice.TableService` method), 126

`update_message()` (`azure.storage.queue.queue.service.QueueService` method), 113

`update_page()` (`azure.storage.blob.pageblobservice.PageBlobService` method), 78

`update_range()` (`azure.storage.file.fileservice.FileService` method), 98

W

`WRITE` (`azure.storage.blob.models.BlobPermissions` attribute), 59

`WRITE` (`azure.storage.blob.models.ContainerPermissions` attribute), 61

`WRITE` (`azure.storage.file.models.FilePermissions` attribute), 101

`WRITE` (`azure.storage.file.models.SharePermissions` attribute), 102

`WRITE` (`azure.storage.models.AccountPermissions` attribute), 130