
tornado-aws

Release 2.0.0

Nov 17, 2019

Contents

1	Clients	3
2	Exceptions	7
3	Examples	9
4	Issues	11
5	Source	13
6	Version History	15
7	Indices and tables	17
	Python Module Index	19
	Index	21

tornado-aws is a low-level Amazon Web Services API client for Tornado.

Contents:

The *AWSClient* and *AsyncAWSClient* implement low-level AWS clients. The clients provide only the mechanism for submitted signed HTTP requests to the AWS APIs and are generally meant to be used by service specific client API implementations.

```
class tornado_aws.client.AWSClient (service, profile=None, region=None, access_key=None,  
secret_key=None, security_token=None, endpoint=None)
```

Implement a low level AWS client that performs the request signing required for AWS API requests.

AWSClient uses the same configuration method and environment variables as the AWS CLI. For configuration information visit the “Getting Set Up” section of the [AWS Command Line Interface user guide](#).

When creating the *AWSClient* instance you need to specify the *service* that you will be interacting with. This value is used when signing the request headers and must match the service values as specified in the [AWS General Reference documentation](#).

The AWS configuration profile can be set when creating the *AWSClient* instance or by setting the `AWS_DEFAULT_PROFILE` environment variable. If neither are set, `default` will be used.

The AWS region is set by reading in configuration or by the `AWS_DEFAULT_REGION` environment variable. If neither or set, it will attempt to be set by invoking the EC2 Instance Metadata and user data API, if available.

The AWS access key can be set when creating a new instance. If it’s not passed in when creating the *AWSClient*, the client will attempt to get the key from the `AWS_ACCESS_KEY_ID` environment variable. If that is not set, it will attempt to get the key from the AWS CLI credentials file. The path to the credentials file can be overridden in the `AWS_SHARED_CREDENTIALS_FILE` environment variable. Note that a value set in `AWS_ACCESS_KEY_ID` will only be used if there is an accompanying value in `AWS_SECRET_ACCESS_KEY` environment variable.

If `AWS_SECURITY_TOKEN` or `AWS_SESSION_TOKEN` are set, they will be automatically be used as well.

Like the access key, the secret key can be set when creating a new client instance. The configuration logic matches the access key with the exception of the environment variable. The secret key can set in the `AWS_SECRET_ACCESS_KEY` environment variable.

If there is no local configuration or credentials, the client will attempt to load the information from the EC2 instance meta-data API, if it is available.

The `endpoint` argument is primarily used for testing and allows for the use of a specified base URL value instead of the auto-construction of a URL using the service and region variables.

Parameters

- **service** (*str*) – The service for the API calls
- **profile** (*str*) – Optionally specify the configuration profile name
- **region** (*str*) – An optional AWS region to make requests to
- **access_key** (*str*) – An optional access key
- **secret_key** (*str*) – An optional secret access key
- **security_token** (*str*) – An optional security token
- **endpoint** (*str*) – Override the base endpoint URL

Raises `tornado_aws.exceptions.ConfigNotFound`

Raises `tornado_aws.exceptions.ConfigParserError`

Raises `tornado_aws.exceptions.NoCredentialsError`

Raises `tornado_aws.exceptions.NoProfileError`

close ()

Closes the underlying HTTP client, freeing any resources used.

fetch (*method*, *path*='/', *query_args*=None, *headers*=None, *body*=b'', *recursed*=False)

Executes a request, returning an `HTTPResponse`.

If an error occurs during the fetch, we raise an `HTTPError` unless the `raise_error` keyword argument is set to `False`.

Parameters

- **method** (*str*) – HTTP request method
- **path** (*str*) – The request path
- **query_args** (*dict*) – Request query arguments
- **headers** (*dict*) – Request headers
- **body** (*bytes*) – The request body
- **recursed** (*bool*) – Internally invoked if it's a recursive fetch

Return type `HTTPResponse`

Raises `HTTPError`

Raises `NoCredentialsError`

Raises `AWSError`

```
class tornado_aws.client.AsyncAWSClient (service, profile=None, region=None, ac-  
cess_key=None, secret_key=None, se-  
curity_token=None, endpoint=None,  
max_clients=100, use_curl=False, io_loop=None,  
force_instance=True)
```

Implement a low level AWS client that performs the request signing required for AWS API requests.

`AsyncClient` uses the same configuration method and environment variables as the AWS CLI. For configuration information visit the “Getting Set Up” section of the [AWS Command Line Interface user guide](#).

When creating the `AWSCli` instance you need to specify the `service` that you will be interacting with. This value is used when signing the request headers and must match the service values as specified in the [AWS General Reference](#) documentation.

The AWS configuration profile can be set when creating the `AWSCli` instance or by setting the `AWS_DEFAULT_PROFILE` environment variable. If neither are set, `default` will be used.

The AWS access key can be set when creating a new instance. If it's not passed in when creating the `AWSCli`, the client will attempt to get the key from the `AWS_ACCESS_KEY_ID` environment variable. If that is not set, it will attempt to get the key from the AWS CLI credentials file. The path to the credentials file can be overridden in the `AWS_SHARED_CREDENTIALS_FILE` environment variable. Note that a value set in `AWS_ACCESS_KEY_ID` will only be used if there is an accompanying value in `AWS_SECRET_ACCESS_KEY` environment variable.

If `AWS_SECURITY_TOKEN` or `AWS_SESSION_TOKEN` are set, they will be automatically be used as well.

Like the access key, the secret key can be set when creating a new client instance. The configuration logic matches the access key with the exception of the environment variable. The secret key can set in the `AWS_SECRET_ACCESS_KEY` environment variable.

The `endpoint` argument is primarily used for testing and allows for the use of a specified base URL value instead of the auto-construction of a URL using the service and region variables.

`max_clients` allows for the specification of the maximum number if concurrent asynchronous HTTP requests that the client will perform.

Parameters

- **service** (*str*) – The service for the API calls
- **profile** (*str*) – Specify the configuration profile name
- **region** (*str*) – The AWS region to make requests to
- **access_key** (*str*) – The access key
- **secret_key** (*str*) – The secret access key
- **security_token** (*str*) – An optional security token
- **endpoint** (*str*) – Override the base endpoint URL
- **max_clients** (*int*) – Max simultaneous HTTP requests (Default: 100)
- **io_loop** (*tornado.ioloop.IOLoop*) – Specify the IOLoop to use
- **force_instance** (*bool*) – Keep an isolated instance of the HTTP client

Raises `tornado_aws.exceptions.ConfigNotFound`

Raises `tornado_aws.exceptions.ConfigParserError`

Raises `tornado_aws.exceptions.NoCredentialsError`

Raises `tornado_aws.exceptions.NoProfileError`

Raises `tornado_aws.exceptions.CurlNotInstalledError`

close ()

Closes the underlying HTTP client, freeing any resources used.

fetch (*method*, *path*='/', *query_args*=None, *headers*=None, *body*=None, *recursed*=False)

Executes a request, returning an `HTTPResponse`.

If an error occurs during the fetch, we raise an `HTTPError` unless the `raise_error` keyword argument is set to `False`.

Parameters

- **method** (*str*) – HTTP request method
- **path** (*str*) – The request path
- **query_args** (*dict*) – Request query arguments
- **headers** (*dict*) – Request headers
- **body** (*bytes*) – The request body
- **recursed** (*bool*) – Internal use only

Return type `HTTPResponse`

Raises `HTTPError`

Raises `AWSError`

Raises `NoCredentialsError`

The following exceptions may be raised during the course of using `tornado_aws.client.AWSClient` and `tornado_aws.client.AsyncAWSClient`:

exception `tornado_aws.exceptions.AWSClientException` (**kwargs)

Base exception class for AWSClient

Variables `msg` – The error message

with_traceback ()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception `tornado_aws.exceptions.AWSError` (**kwargs)

Raised when the credentials could not be located.

with_traceback ()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception `tornado_aws.exceptions.ConfigNotFound` (**kwargs)

The configuration file could not be parsed.

Variables `path` – The path to the config file

with_traceback ()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception `tornado_aws.exceptions.ConfigParserError` (**kwargs)

Error raised when parsing a configuration file with `:py:class'configparser.RawConfigParser'`

Variables `path` – The path to the config file

with_traceback ()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception `tornado_aws.exceptions.CurlNotInstalledError`

Raised when the `CurlAsyncHTTPClient` is requested but `pycurl` is not installed.

with_traceback ()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception `tornado_aws.exceptions.LocalCredentialsError (**kwargs)`
Raised when the credentials could not be located.

with_traceback()
Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception `tornado_aws.exceptions.NoCredentialsError (**kwargs)`
Raised when the credentials could not be located.

with_traceback()
Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception `tornado_aws.exceptions.NoProfileError (**kwargs)`
Raised when the specified profile could not be located.

Variables

- **path** – The path to the config file
- **profile** – The profile that was specified

with_traceback()
Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception `tornado_aws.exceptions.RequestException (**kwargs)`
Raised when a request failed due to a network issue.

Variables error – The error which occurred

with_traceback()
Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

The following example uses invokes the DescribeTable endpoint for DynamoDB:

```
import json
import pprint

import tornado_aws
from tornado import gen, ioloop

HEADERS = {'Content-Type': 'application/x-amz-json-1.0',
           'x-amz-target': 'DynamoDB_20120810.DescribeTable'}
PAYLOAD = {'TableName': 'prod-us-east-1-history-events'}

@gen.coroutine
def async_request():
    client = tornado_aws.AsyncAWSClient('dynamodb')
    response = yield client.fetch('POST', '/', headers=HEADERS,
                                 body=json.dumps(PAYLOAD))
    x = json.loads(response.body.decode('utf-8'))
    pprint.pprint(x)
    ioloop.IOLoop.instance().stop()

_ioloop = ioloop.IOLoop.instance()
_ioloop.add_callback(async_request)
_ioloop.start()
```


CHAPTER 4

Issues

Please report any issues to the Github repo at <https://github.com/gmr/tornado-aws/issues>

CHAPTER 5

Source

tornado-aws source is available on Github at <https://github.com/gmr/tornado-aws>

CHAPTER 6

Version History

See history

CHAPTER 7

Indices and tables

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

t

`tornado_aws.client`, 3

`tornado_aws.exceptions`, 7

A

AsyncAWSClient (*class in tornado_aws.client*), 4
 AWSClient (*class in tornado_aws.client*), 3
 AWSClientException, 7
 AWSError, 7

C

close() (*tornado_aws.client.AsyncAWSClient method*), 5
 close() (*tornado_aws.client.AWSClient method*), 4
 ConfigNotFound, 7
 ConfigParserError, 7
 CurlNotInstalledError, 7

F

fetch() (*tornado_aws.client.AsyncAWSClient method*), 5
 fetch() (*tornado_aws.client.AWSClient method*), 4

L

LocalCredentialsError, 7

N

NoCredentialsError, 8
 NoProfileError, 8

R

RequestException, 8

T

tornado_aws.client (*module*), 3
 tornado_aws.exceptions (*module*), 7

W

with_traceback() (*tornado_aws.exceptions.AWSClientException method*), 7
 with_traceback() (*tornado_aws.exceptions.AWSError method*), 7

with_traceback() (*tornado_aws.exceptions.ConfigNotFound method*), 7

with_traceback() (*tornado_aws.exceptions.ConfigParserError method*), 7

with_traceback() (*tornado_aws.exceptions.CurlNotInstalledError method*), 7

with_traceback() (*tornado_aws.exceptions.LocalCredentialsError method*), 8

with_traceback() (*tornado_aws.exceptions.NoCredentialsError method*), 8

with_traceback() (*tornado_aws.exceptions.NoProfileError method*), 8

with_traceback() (*tornado_aws.exceptions.RequestException method*), 8