

---

# Shopify Requests

*Release v0.3.0*

**Paul Robertson**

**Apr 13, 2019**



## API DOCS:

<b>1</b>	<b>User's Guide</b>	<b>3</b>
<b>2</b>	<b>API Docs</b>	<b>5</b>
<b>3</b>	<b>Indices and tables</b>	<b>11</b>
	<b>Python Module Index</b>	<b>13</b>



The ShopifyRequests library is a wrapper around the python requests library. Its main purpose is to remove the boiler plate code needed to do basic API calls to Shopify.



## USER'S GUIDE

The *RestClient* is meant to be as simple as using the requests library. All the http verbs needed for REST calls are available.

- GET
- PUT
- POST
- PATCH
- DELETE

OAuth token example:

```
from shopify_requests import RestClient
client = RestClient(
    'foo.myshopify.com',
    access_token='abc123',
)
response = client.get('shop.json')
```

Private app example:

```
from shopify_requests import RestClient
client = RestClient(
    'foo.myshopify.com',
    username='1234',
    password='asdf',
)
response = client.get('shop.json')
```

By default the client also has safe retries enabled to help with network issues. If a request fails to send data to Shopify, it will be retried a 3 times before returning the failure response. This can be configured with the *connect\_retries* parameter.

```
from shopify_requests import RestClient
client = RestClient(
    'foo.myshopify.com',
    access_token='abc123',
    connect_retries=5,
)
response = client.get('shop.json')
```

Reusing the client has the added benefit of reusing the http session once the connection is established. This means that subsequent calls will not have to do the SSL handshake.





## 2.1 shopify\_requests

### 2.1.1 shopify\_requests package

#### Submodules

#### shopify\_requests.rest\_client module

```
class shopify_requests.rest_client.RestClient (myshopify_domain, access_token=None,  
                                              username=None, password=None, connect_retries=None,  
                                              backoff_factor=None, max_limit_retries=None,  
                                              limit_backoff_factor=None, version=None)
```

Bases: object

Create a session client for this Shopify store.

#### Note

- Only *access\_token* or (*username* and *password*) are required
- **urllib3** documents *backoff\_factor* as
  - $\{\text{backoff factor}\} * (2 ** (\{\text{number of total retries}\} - 1))$

#### Parameters

- **myshopify\_domain** – foobar.myshopify.com
- **access\_token** – OAuth token from a public app
- **username** – Username for private app
- **password** – Password for private app
- **connect\_retries** – [default: 3] how many connection-related errors to retry on
- **backoff\_factor** – [default: 0.5] how quickly to backoff for safe retries
- **max\_limit\_retries** – [default: 0] how many retries to do when rate limited
- **limit\_backoff\_factor** – [default: 0.5] same as *backoff\_factor* but for rate limited
- **version** – [default: unstable] the Shopify API version to use

**calls\_remaining()**

Return the number of API calls remaining.

**Return type** int

**See also** [Shopify API call limits](#)

**calls\_used()**

Return the number of API calls used.

**Return type** int

**See also** [Shopify API call limits](#)

**close()**

Close the session.

**See also** [requests.Session.close](#).

**delete(url, \*\*kwargs)**

Send a DELETE request.

**Parameters**

- **url** – URL for the new Request object.
- **\*\*kwargs** – Optional arguments that session takes.

**Return type** [requests.Response](#)

**See also** [requests.Session.delete](#).

**get(url, \*\*kwargs)**

Send a GET request.

**Parameters**

- **url** – URL for the new Request object.
- **\*\*kwargs** – Optional arguments that session takes.

**Return type** [requests.Response](#)

**See also** [requests.Session.get](#).

**max\_available()**

Return the bucket size for the token.

**Return type** int

**See also** [Shopify API call limits](#)

**patch(url, \*\*kwargs)**

Send a PATCH request.

**Parameters**

- **url** – URL for the new Request object.
- **\*\*kwargs** – Optional arguments that session takes.

**Return type** [requests.Response](#)

**See also** [requests.Session.patch](#).

**post(url, \*\*kwargs)**

Send a POST request.

**Parameters**

- **url** – URL for the new Request object.
- **\*\*kwargs** – Optional arguments that session takes.

**Return type** `requests.Response`

**See also** `requests.Session.post`.

**put** (*url*, *\*\*kwargs*)

Send a PUT request.

#### Parameters

- **url** – URL for the new Request object.
- **\*\*kwargs** – Optional arguments that session takes.

**Return type** `requests.Response`

**See also** `requests.Session.put`.

## Module contents

```
class shopify_requests.RestClient (myshopify_domain, access_token=None, user-  
name=None, password=None, connect_retries=None,  
backoff_factor=None, max_limit_retries=None,  
limit_backoff_factor=None, version=None)
```

Bases: `object`

Create a session client for this Shopify store.

#### Note

- Only *access\_token* or (*username* and *password*) are required
- `urllib3` documents *backoff\_factor* as  
– `{backoff factor} * (2 ** ({number of total retries} - 1))`

#### Parameters

- **myshopify\_domain** – foobar.myshopify.com
- **access\_token** – OAuth token from a public app
- **username** – Username for private app
- **password** – Password for private app
- **connect\_retries** – [default: 3] how many connection-related errors to retry on
- **backoff\_factor** – [default: 0.5] how quickly to backoff for safe retries
- **max\_limit\_retries** – [default: 0] how many retries to do when rate limited
- **limit\_backoff\_factor** – [default: 0.5] same as *backoff\_factor* but for rate limited
- **version** – [default: unstable] the Shopify API version to use

**calls\_remaining** ()

Return the number of API calls remaining.

**Return type** `int`

See also [Shopify API call limits](#)

**calls\_used()**

Return the number of API calls used.

**Return type** `int`

See also [Shopify API call limits](#)

**close()**

Close the session.

See also [requests.Session.close](#).

**delete(url, \*\*kwargs)**

Send a DELETE request.

**Parameters**

- **url** – URL for the new Request object.
- **\*\*kwargs** – Optional arguments that session takes.

**Return type** `requests.Response`

See also [requests.Session.delete](#).

**get(url, \*\*kwargs)**

Send a GET request.

**Parameters**

- **url** – URL for the new Request object.
- **\*\*kwargs** – Optional arguments that session takes.

**Return type** `requests.Response`

See also [requests.Session.get](#).

**max\_available()**

Return the bucket size for the token.

**Return type** `int`

See also [Shopify API call limits](#)

**patch(url, \*\*kwargs)**

Send a PATCH request.

**Parameters**

- **url** – URL for the new Request object.
- **\*\*kwargs** – Optional arguments that session takes.

**Return type** `requests.Response`

See also [requests.Session.patch](#).

**post(url, \*\*kwargs)**

Send a POST request.

**Parameters**

- **url** – URL for the new Request object.
- **\*\*kwargs** – Optional arguments that session takes.

**Return type** [requests.Response](#)

**See also** [requests.Session.post](#).

**put** (*url*, *\*\*kwargs*)

Send a PUT request.

**Parameters**

- **url** – URL for the new Request object.
- **\*\*kwargs** – Optional arguments that session takes.

**Return type** [requests.Response](#)

**See also** [requests.Session.put](#).



## INDICES AND TABLES

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





## PYTHON MODULE INDEX

### S

`shopify_requests`, [7](#)

`shopify_requests.rest_client`, [5](#)



## INDEX

### C

`calls_remaining()`  
    (*shopify\_requests.rest\_client.RestClient*  
        *method*), 6  
`calls_remaining()` (*shopify\_requests.RestClient*  
    *method*), 7  
`calls_used()` (*shopify\_requests.rest\_client.RestClient*  
    *method*), 6  
`calls_used()` (*shopify\_requests.RestClient method*),  
    8  
`close()` (*shopify\_requests.rest\_client.RestClient*  
    *method*), 6  
`close()` (*shopify\_requests.RestClient method*), 8

### D

`delete()` (*shopify\_requests.rest\_client.RestClient*  
    *method*), 6  
`delete()` (*shopify\_requests.RestClient method*), 8

### G

`get()` (*shopify\_requests.rest\_client.RestClient method*),  
    6  
`get()` (*shopify\_requests.RestClient method*), 8

### M

`max_available()` (*shopify\_requests.rest\_client.RestClient*  
    *method*), 6  
`max_available()` (*shopify\_requests.RestClient*  
    *method*), 8

### P

`patch()` (*shopify\_requests.rest\_client.RestClient*  
    *method*), 6  
`patch()` (*shopify\_requests.RestClient method*), 8  
`post()` (*shopify\_requests.rest\_client.RestClient*  
    *method*), 6  
`post()` (*shopify\_requests.RestClient method*), 8  
`put()` (*shopify\_requests.rest\_client.RestClient method*),  
    7  
`put()` (*shopify\_requests.RestClient method*), 9

### R

*RestClient (class in shopify\_requests)*, 7  
*RestClient (class in shopify\_requests.rest\_client)*, 5

### S

*shopify\_requests (module)*, 7  
*shopify\_requests.rest\_client (module)*, 5