

---

# **pubg***clientDocumentation*

***Release 0.1.4***

**Hunter Senft-Grupp**

**Mar 13, 2018**



---

## Contents:

---

<b>1</b>	<b>Installing</b>	<b>3</b>
<b>2</b>	<b>Development</b>	<b>5</b>
<b>3</b>	<b>Examples</b>	<b>7</b>
3.1	Get Status . . . . .	7
3.2	Get Matches . . . . .	7
3.3	Add Search Options . . . . .	8
<b>4</b>	<b>Development Guides</b>	<b>9</b>
4.1	Adding a New Model . . . . .	9
<b>5</b>	<b>API</b>	<b>11</b>
5.1	pubg_client . . . . .	11
5.2	client . . . . .	11
5.3	api . . . . .	13
5.4	endpoints . . . . .	13
5.5	models . . . . .	14
<b>6</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>



## A python client for the pubg API

Features:

- Versioned API Endpoint and response payloads
- Pagination Support

**Danger:** This library is in alpha. I will try my utmost to not make backwards incompatible changes but it is possible that they will be necessary.



# CHAPTER 1

---

## Installing

---

```
$ pip install pubg-client
```





## CHAPTER 2

---

### Development

---

```
$ git clone https://github.com/huntcsg/pubg-client.git
$ cd pubg-client
$ ./utils/manage clean
$ ./utils/manage test
$ ./utils/manage docs
```

1. All pull requests must pass the travis-ci builds
2. All pull requests should include inline (docstring) documentation, updates to built documentation if applicable, and test coverage. This project aspires to be a 100% test coverage library.



### 3.1 Get Status

```
from pubg_client import Client

client = Client(autocall=True)

status = client.api.status()

print(status)
print(status.version)
print(status.released_at)
```

### 3.2 Get Matches

Get some matches:

```
from pubg_client import Client
import os

client = Client(token=os.environ['PUBG_API_KEY'], autocall=True)
matches = client.api.matches()
```

Get the next matches:

```
from pubg_client import Client
import os

client = Client(token=os.environ['PUBG_API_KEY'], autocall=True)
matches = client.api.matches()

next_matches = matches.next(client)
```

## 3.3 Add Search Options

Do some sorting

```
from pubg_client import Client
import os

client = Client(token=os.environ['PUBG_API_KEY'], autocall=False)
prepared_request = client.api.matches().sort('createdAt')
matches = prepared_request()
```

Filter the matches (and sort)

```
from pubg_client import Client
import os

client = Client(token=os.environ['PUBG_API_KEY'], autocall=False)
prepared_request = client.api.matches().sort('createdAt').filter('createdAt',
↪ '2018-04-01T00:01:01Z')
matches = prepared_request()
```

## 4.1 Adding a New Model

1. Find or create a new python module under *src/pubg\_client/models/*

```
from ._base import Mapping, Model

class NewModel(Model):

    key_map = {
        'key': Mapping('key_in_api', type)
    }
```

2. If needed, define a *load*, *next*, and *previous* method.
3. If appropriate, register it as the deserializer for an endpoint (or in another models key map)



## 5.1 pubg\_client

Pubg Client package docstring

## 5.2 client

Client Module

```
class pubg_client._client.Client (base_url='https://api.playbattlegrounds.com', token=None,  
                                shard='pc-na', raw=False, autocall=False)
```

The client class

**api**

**Returns**

**current\_page** (*response*)

**Parameters response** –

**Returns**

**get\_shard** (*shard=None*)

**Parameters shard** –

**Returns**

**get\_token** (*token=None*)

**Parameters token** –

**Returns**

**make\_request** (*request*)

**Parameters request** –

**Returns**

**next\_page** (*response*)

**Parameters** *response* –

**Returns**

**prepare\_request** (*method, url, gzip=False, \*\*kwargs*)

**Parameters**

- *method* –
- *url* –
- *gzip* –
- *kwargs* –

**Returns**

**previous\_page** (*response*)

**Parameters** *response* –

**Returns**

**request** (*method, url, \*\*kwargs*)

**Parameters**

- *method* –
- *url* –
- *kwargs* –

**Returns**

**validate\_shard** (*shard*)

**Parameters** *shard* –

**Returns**

**class** `pubg_client._client.PUBGRequest` (*prepared\_request, client\_or\_endpoint*)

A lazy request object

**filter** (*filter\_name, filter\_value*)

**Parameters**

- *filter\_name* –
- *filter\_value* –

**Returns**

**limit** (*limit*)

**Parameters** *limit* –

**Returns**

**offset** (*offset*)

**Parameters** *offset* –

**Returns**



**sort** (*sort\_key*)

Parameters **sort\_key** –

Returns

## 5.3 api

**class** `pubg_client._api.API` (*client*, *raw=False*, *autocall=False*)

The api namespace

**match**

**matches**

Matches Endpoint

**classmethod** **register** (*name*, *deserializer*)

Parameters

- **name** –
- **deserializer** –

Returns

**classmethod** **register\_endpoint** (*name*, *deserializer*, *endpoint*)

Parameters

- **name** –
- **deserializer** –
- **endpoint** –

Returns

**status**

The status endpoint

## 5.4 endpoints

**class** `pubg_client.endpoints.Match` (*deserializer=None*, *client=None*)

**method** = 'GET'

**path** = 'matches/{id}'

**requires\_shard** = True

**url** (*id*)

Parameters **id** –

Returns

**class** `pubg_client.endpoints.Matches` (*deserializer=None*, *client=None*)

Matches Endpoint

**method** = 'GET'

```
    path = 'matches'
    requires_shard = True

class pubg_client.endpoints.Status (deserializer=None, client=None)
    The status endpoint

    method = 'GET'

    path = 'status'
```

## 5.5 models

```
class pubg_client.models.Match (**kwargs)
    A match object

    key_map = {'assets': Mapping(pubg_key='assets', cls=<pubg_client.models._base.List ob

class pubg_client.models.MatchCollection (**kwargs)
    A collection of matches with navigation convenience methods

    classmethod load (payload)
        Parameters payload –
        Returns

    next ()

class pubg_client.models.Status (**kwargs)
    A status

    key_map = {'attributes': Mapping(pubg_key='attributes', cls=StatusAttributes('version

    released_at
        Convenience Method

    version
        Convenience Method

class pubg_client.models.StatusAttributes (**kwargs)
    A status attributes object. This is so that we can parse the released at string into a Datetime

    key_map = {'released_at': Mapping(pubg_key='releasedAt', cls=<function parse at 0x7ff

class pubg_client.models.Participant (**kwargs)
    A participant object

    key_map = {'actor': Mapping(pubg_key='actor', cls=<class 'str'>), 'id': Mapping(pubg

class pubg_client.models.Asset (**kwargs)
    Asset objects contain a URL string that links to a telemetry.json file, which will contain an array of event objects
    that provide further insight into a match.

    key_map = {'content_type': Mapping(pubg_key='contentType', cls=<class 'str'>), 'creat

class pubg_client.models.AssetCollection (**kwargs)
    A collection of assets

    classmethod load (payload)
        Parameters payload –
        Returns
```

```
class pubg_client.models.Roster(**kwargs)
    A roster object

    key_map = {'id': Mapping(pubg_key='id', cls=<class 'str'>), 'participants': Mapping()
```



## CHAPTER 6

---

### Indices and tables

---

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



### p

- `pubg_client`, [11](#)
- `pubg_client._api`, [13](#)
- `pubg_client._client`, [11](#)
- `pubg_client.endpoints`, [13](#)
- `pubg_client.models`, [14](#)





### P

`pubg_client` (module), 11  
`pubg_client._api` (module), 13  
`pubg_client._client` (module), 11  
`pubg_client.endpoints` (module), 13  
`pubg_client.models` (module), 14