
clashroyale Documentation

Release 4.0.0

fourjr, arq

Nov 06, 2019

Contents

1	Features	3
2	Contents	5
2.1	API Reference	5
2.1.1	OfficialAPI	5
2.1.2	RoyaleAPI	12
2.1.3	Exceptions	23
2.2	Setting Up Logging	24
3	Indices	25
	Index	27

The clashroyale module is an easy to use, feature-rich and async ready API wrapper for both the [Official API](#) and [RoyaleAPI](#)

CHAPTER 1

Features

- Use the same client for sync/async operations.
- Implements the entire API.
- Easy to use with an object oriented design.
- Optional caching for reliability and performance.
- Continously maintained

2.1 API Reference

The following section outlines the API of :clashroyale: and how to access the Official API and the unofficial RoyaleAPI.

2.1.1 OfficialAPI

class `clashroyale.official_api.Client` (*token*, *session=None*, *is_async=False*, ***options*)

A client that requests data from `api.clashroyale.com`. This class can either be async or non async.

Parameters

- **token** (*str*) – The api authorization token to be used for requests. <https://developer.clashroyale.com/>
- **is_async** (*Optional[bool]* = *False*) – Toggle for asynchronous/synchronous usage of the client
- **error_debug** (*Optional[bool]* = *False*) – Toggle for every method to raise `ServerError` to test error handling.
- **session** (*Optional[Session]* = *None*) – The http (client)session to be used for requests. Can either be a `requests.Session` or `aiohttp.ClientSession`.
- **timeout** (*Optional[int]* = *10*) – A timeout for requests to the API
- **url** (*Optional[str]* = `'https://api.clashroyale.com/v1'`) – A url to use instead of `api.clashroyale.com/v1` Only use this if you know what you are doing.
- **cache_fp** (*Optional[str]* = *None*) – File path for the sqlite3 database to use for caching requests, if this parameter is provided, the client will use its caching system
- **cache_expires** (*Optional[int]* = *10*) – The number of seconds to wait before the client will request from the api for a specific route

- **table_name** (*Optional[str] = 'cache'*) – The table name to use for the cache database.
- **camel_case** (*Optional[bool] = False*) – Whether or not to access model data keys in snake_case or camelCase, this defaults to use snake_case
- **constants** (*Optional[dict] = None*) – Constants to use instead of the ones updated when the package is re-installed. To extract a dict from a BaseAttrDict, do `BaseAttrDict.to_dict()`
- **user_agent** (*Optional[str] = None*) – Appends to the default user-agent

classmethod Async (*token, session=None, **options*)

Returns the client in async mode.

get_all_cards (*timeout: int = None*)

Get a list of all the cards in the game

Parameters timeout (*Optional[int] = None*) – Custom timeout that overwrites `Client.timeout`

get_all_locations (*timeout: int = None*)

Get a list of all locations

Parameters timeout (*Optional[int] = None*) – Custom timeout that overwrites `Client.timeout`

get_arena_image (*obj: clashroyale.official_api.models.BaseAttrDict*)

Get the arena image URL

Parameters

- **obj** (*official_api.models.BaseAttrDict*) – An object that has the arena ID in `.arena.id` Can be `Profile` for example.
- **None or str** (*Returns*) –

get_card_info (*card_name: str*)

Returns card info from constants

Parameters

- **card_name** (*str*) – A card name
- **None or Constants** (*Returns*) –

get_clan (*tag: clashroyale.official_api.utils.crtag, timeout: int = None*)

Get information about a clan

Parameters

- **tag** (*str*) – A valid tournament tag. Minimum length: 3 Valid characters: 0289PYLQ-GRJCUV
- **timeout** (*Optional[int] = None*) – Custom timeout that overwrites `Client.timeout`

get_clan_image (*obj: clashroyale.official_api.models.BaseAttrDict*)

Get the clan badge image URL

Parameters

- **obj** (*official_api.models.BaseAttrDict*) – An object that has the clan badge ID either in `.clan.badge_id` or `.badge_id` Can be a clan or a profile for example.
- **str** (*Returns*) –

get_clan_members (*tag: clashroyale.official_api.utils.crtag, **params*)

Get the clan's members

Parameters

- **tag** (*str*) – A valid tournament tag. Minimum length: 3 Valid characters: 0289PYLQ-GRJCUV
- ****limit** (*Optional[int] = None*) – Limit the number of items returned in the response
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites `Client.timeout`

get_clan_war (*tag: clashroyale.official_api.utils.crtag, timeout: int = None*)

Get information about a clan's current clan war

Parameters

- **tag** (*str*) – A valid tournament tag. Minimum length: 3 Valid characters: 0289PYLQ-GRJCUV
- **timeout** (*Optional[int] = None*) – Custom timeout that overwrites `Client.timeout`

get_clan_war_log (*tag: clashroyale.official_api.utils.crtag, **params*)

Get a clan's war log

Parameters

- **tag** (*str*) – A valid tournament tag. Minimum length: 3 Valid characters: 0289PYLQ-GRJCUV
- ****limit** (*Optional[int] = None*) – Limit the number of items returned in the response
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites `Client.timeout`

get_datetime (*timestamp: str, unix=True*)

Converts a `%Y%m%dT%H%M%S.%fZ` to a UNIX timestamp or a `datetime.datetime` object

Parameters

- **timestamp** (*str*) – A timestamp in the `%Y%m%dT%H%M%S.%fZ` format, usually returned by the API in the `created_time` field for example (eg. `20180718T145906.000Z`)
- **unix** (*Optional[bool] = True*) – Whether to return a POSIX timestamp (seconds since epoch) or not
- **int or datetime.datetime** (*Returns*) –

get_deck_link (*deck: clashroyale.official_api.models.BaseAttrDict*)

Form a deck link

Parameters

- **deck** (*official_api.models.BaseAttrDict*) – An object is a deck. Can be retrieved from `Player.current_deck`
- **str** (*Returns*) –

get_location (*location_id: int, timeout: int = None*)

Get a location information

Parameters

- **location_id** (*int*) – A location ID See <https://github.com/RoyaleAPI/cr-api-data/blob/master/json/regions.json> for a list of acceptable location IDs
- **timeout** (*Optional[int] = None*) – Custom timeout that overwrites Client.timeout

get_player (*tag: clashroyale.official_api.utils.crtag, timeout=None*)

Get information about a player

Parameters

- **tag** (*str*) – A valid tournament tag. Minimum length: 3 Valid characters: 0289PYLQ-GRJCUV
- **timeout** (*Optional[int] = None*) – Custom timeout that overwrites Client.timeout

get_player_battles (*tag: clashroyale.official_api.utils.crtag, **params*)

Get a player's battle log

Parameters

- **tag** (*str*) – A valid tournament tag. Minimum length: 3 Valid characters: 0289PYLQ-GRJCUV
- ****limit** (*Optional[int] = None*) – Limit the number of items returned in the response
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites Client.timeout

get_player_chests (*tag: clashroyale.official_api.utils.crtag, timeout: int = None*)

Get information about a player's chest cycle

Parameters

- **tag** (*str*) – A valid tournament tag. Minimum length: 3 Valid characters: 0289PYLQ-GRJCUV
- **timeout** (*Optional[int] = None*) – Custom timeout that overwrites Client.timeout

get_player_verify (*tag: clashroyale.official_api.utils.crtag, apikey: str, timeout=None*)

Check the API Key of a player. This endpoint has been **restricted** to certain members of the community

Raises BadRequest if the apikey is invalid

Parameters

- **tag** (*str*) – A valid tournament tag. Minimum length: 3 Valid characters: 0289PYLQ-GRJCUV
- **apikey** (*str*) – The API Key in the player's settings
- **timeout** (*Optional[int] = None*) – Custom timeout that overwrites Client.timeout

get_rarity_info (*rarity: str*)

Returns card info from constants

Parameters

- **rarity** (*str*) – A rarity name

- **None or Constants** (*Returns*) –

get_top_clans (*location_id='global', **params*)

Get a list of top clans by trophy

Parameters

- **location_id** (*Optional[str] = 'global'*) – A location ID or global See <https://github.com/RoyaleAPI/cr-api-data/blob/master/json/regions.json> for a list of acceptable location IDs
- ****limit** (*Optional[int] = None*) – Limit the number of items returned in the response
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites `Client.timeout`

get_top_clanwar_clans (*location_id='global', **params*)

Get a list of top clan war clans

Parameters

- **location_id** (*Optional[str] = 'global'*) – A location ID or global See <https://github.com/RoyaleAPI/cr-api-data/blob/master/json/regions.json> for a list of acceptable location IDs
- ****limit** (*Optional[int] = None*) – Limit the number of items returned in the response
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites `Client.timeout`

get_top_players (*location_id='global', **params*)

Get a list of top players

Parameters

- **location_id** (*Optional[str] = 'global'*) – A location ID or global See <https://github.com/RoyaleAPI/cr-api-data/blob/master/json/regions.json> for a list of acceptable location IDs
- ****limit** (*Optional[int] = None*) – Limit the number of items returned in the response
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites `Client.timeout`

get_tournament (*tag: clashroyale.official_api.utils.crtag, timeout=0*)

Get a tournament information

Parameters

- **tag** (*str*) – A valid tournament tag. Minimum length: 3 Valid characters: 0289PYLQ-GRJCUV
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites `Client.timeout`

search_clans (***params*)

Search for a clan. At least one of the filters must be present

Parameters

- **name** (*Optional[str]*) – The name of a clan (has to be at least 3 characters long)

- **locationId** (*Optional[int]*) – A location ID
- **minMembers** (*Optional[int]*) – The minimum member count of a clan
- **maxMembers** (*Optional[int]*) – The maximum member count of a clan
- **minScore** (*Optional[int]*) – The minimum trophy score of a clan
- ****limit** (*Optional[int] = None*) – Limit the number of items returned in the response
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites `Client.timeout`

search_tournaments (*name: str, **params*)

Search for a tournament by its name

Parameters

- **name** (*str*) – The name of a tournament
- ****limit** (*Optional[int] = None*) – Limit the number of items returned in the response
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites `Client.timeout`

Data Models

class `clashroyale.official_api.models.BaseAttrDict` (*client, data, response, cached=False, ts=None*)

This class is the base class for all models, its a wrapper around the [python-box](#) which allows access to data via dot notation, in this case, API data will be accessed using this class. This class shouldnt normally be used by the user since its a base class for the actual models returned from the client.

Example

Accessing data via dot notation:

This functionality allows this library to present API data in a clean dynamic way.

raw_data

The raw data in the form of a dictionary being used

Type dict

cached

Whether or not the data being used is cached data from the cache database.

Type bool

last_updated

When the data which is currently being used was last updated.

Type datetime.datetime

response

Response object containing headers and more information. Returns None if cached

Type requests.Response or aiohttp.ClientResponse or None

```
class clashroyale.official_api.models.Refreshable (client, data, response,  
                                                cached=False, ts=None)
```

Mixin class for re requesting data from the api for the specific model.

```
refresh ()
```

(a)sync refresh the data.

```
class clashroyale.official_api.models.PaginatedAttrDict (client, data, response,  
                                                         model, cached=False,  
                                                         ts=None)
```

Mixin class to allow for the paginated endpoints to be iterable

Example

Searching clans with a limit of 3:

This functionality allows this library to present API data in a clean dynamic way.

Best use case: Set the `limit` to as low as possible without compromising runtime. Everytime the `limit` has been hit, an API call is made.

```
all_data ()
```

Loops through and adds all data to the `raw_data`

This has a chance to get 429 RateLimitError

```
update_data ()
```

Adds the NEXT data in the `raw_data` dictionary. Returns True if data is added. Returns False if data is not added

```
class clashroyale.official_api.models.PartialClan (client, data, response,  
                                                    cached=False, ts=None)
```

```
get_clan ()
```

(a)sync function to return clan.

```
class clashroyale.official_api.models.PartialPlayer (client, data, response,  
                                                       cached=False, ts=None)
```

```
get_player ()
```

(a)sync function to return player.

```
class clashroyale.official_api.models.PartialPlayerClan (client, data, response,  
                                                           cached=False, ts=None)
```

Brief player model, does not contain full data, non refreshable.

```
get_clan ()
```

(a)sync function to return clan.

```
get_player ()
```

(a)sync function to return player.

```
class clashroyale.official_api.models.Member (clan, data, response)
```

A clan member model, keeps a reference to the clan object it came from.

```
get_player ()
```

(a)sync function to return player.

```
class clashroyale.official_api.models.FullPlayer (client, data, response, cached=False,  
                                                    ts=None)
```

A clash royale player model.

get_clan()
(a)sync function to return clan.

refresh()
(a)sync refresh the data.

class clashroyale.official_api.models.**FullClan**(*client, data, response, cached=False, ts=None*)

A clash royale clan model, full data + refreshable.

refresh()
(a)sync refresh the data.

class clashroyale.official_api.models.**rlist**(*client, data, cached, ts, response*)

append()
Append object to the end of the list.

clear()
Remove all items from list.

copy()
Return a shallow copy of the list.

count()
Return number of occurrences of value.

extend()
Extend list by appending elements from the iterable.

index()
Return first index of value.

Raises ValueError if the value is not present.

insert()
Insert object before index.

pop()
Remove and return item at index (default last).

Raises IndexError if list is empty or index is out of range.

refresh()
(a)sync refresh the data.

remove()
Remove first occurrence of value.

Raises ValueError if the value is not present.

reverse()
Reverse *IN PLACE*.

sort()
Stable sort *IN PLACE*.

2.1.2 RoyaleAPI

class clashroyale.royaleapi.**Client**(*token, session=None, is_async=False, **options*)
A client that requests data from royaleapi.com. This class can either be async or non async.

Parameters

- **token** (*str*) – The api authorization token to be used for requests <https://docs.royaleapi.com/#/authentication>
- **is_async** (*Optional[bool]* = *False*) – Toggle for asynchronous/synchronous usage of the client
- **error_debug** (*Optional[bool]* = *False*) – Toggle for every method to raise `ServerError` to test error handling
- **session** (*Optional[Session]* = *None*) – The http (client)session to be used for requests. Can either be a `requests.Session` or `aiohttp.ClientSession`
- **timeout** (*Optional[int]* = *10*) – A timeout for requests to the API
- **url** (*Optional[str]* = <https://api.royaleapi.com>) – A url to use instead of `api.royaleapi.com`. Only use this if you know what you are doing
- **cache_fp** (*Optional[str]*) – File path for the sqlite3 database to use for caching requests, if this parameter is provided, the client will use its caching system
- **cache_expires** (*Optional[int]* = *10*) – The number of seconds to wait before the client will request from the api for a specific route
- **table_name** (*Optional[str]* = *'cache'*) – The table name to use for the cache database
- **camel_case** (*Optional[bool]* = *False*) – Whether or not to access model data keys in `snake_case` or `camelCase`, this defaults use `snake_case`
- **user_agent** (*Optional[str]* = *None*) – Appends to the default user-agent

classmethod Async (*token, session=None, **options*)

Returns the client in async mode.

get_1k_tournaments (***params*)

Get a list of tournaments that have at least 1000 max players

****open: Optional[int] = 0** Set to 1 to filter tournaments that are open

****full: Optional[int] = 0** Set to 1 to filter tournaments that are full

****inprep: Optional[int] = 0** Set to 1 to filter tournaments that are in preperation

****joinable: Optional[int] = 0** Set to 1 to filter tournaments that are joinable

****keys: Optional[list] = None** Filter which keys should be included in the response

****exclude: Optional[list] = None** Filter which keys should be excluded from the response

****max: Optional[int] = None** Limit the number of items returned in the response

****page: Optional[int] = None** Works with max, the zero-based page of the items

****timeout: Optional[int] = None** Custom timeout that overwrites `Client.timeout`

get_clan (**tags, **params*)

Get a clan information

Parameters

- ***tags** (*str*) – Valid clan tags. Minimum length: 3 Valid characters: 0289PYLQGR-JCUV
- ****keys** (*Optional[list]* = *None*) – Filter which keys should be included in the response

- ****exclude** (*Optional[list] = None*) – Filter which keys should be excluded from the response
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites Client.timeout

get_clan_battles (*tags, **params)

Get the battle log from everyone in the clan

Parameters

- ***tags** (*str*) – Valid player tags. Minimum length: 3 Valid characters: 0289PYLQGR-JCUV
- ****type** (*str*) – Filters what kind of battles. Pick from: :all:, :war:, :clanMate:
- ****keys** (*Optional[list] = None*) – Filter which keys should be included in the response
- ****exclude** (*Optional[list] = None*) – Filter which keys should be excluded from the response
- ****max** (*Optional[int] = None*) – Limit the number of items returned in the response
- ****page** (*Optional[int] = None*) – Works with max, the zero-based page of the items
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites Client.timeout

get_clan_history (*tags, **params)

Get the clan history. Only works if the clan is being tracked by having either cr-api.com or royaleapi.com in the clan's description

Parameters

- ***tags** (*str*) – Valid clan tags. Minimum length: 3 Valid characters: 0289PYLQGR-JCUV
- ****keys** (*Optional[list] = None*) – Filter which keys should be included in the response
- ****exclude** (*Optional[list] = None*) – Filter which keys should be excluded from the response
- ****max** (*Optional[int] = None*) – Limit the number of items returned in the response
- ****page** (*Optional[int] = None*) – Works with max, the zero-based page of the items
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites Client.timeout

get_clan_tracking (*tags, **params)

Returns if the clan is currently being tracked by the API by having either cr-api.com or royaleapi.com in the clan description

Parameters

- ***tags** (*str*) – Valid clan tags. Minimum length: 3 Valid characters: 0289PYLQGR-JCUV

- ****keys** (*Optional[list] = None*) – Filter which keys should be included in the response
- ****exclude** (*Optional[list] = None*) – Filter which keys should be excluded from the response
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites `Client.timeout`

get_clan_war (*tag: clashroyale.royaleapi.utils.crtag, **params*)

Get information about a clan's current clan war

Parameters

- ***tag** (*str*) – A valid clan tag. Minimum length: 3 Valid characters: 0289PYLQGR-JCUV
- ****keys** (*Optional[list] = None*) – Filter which keys should be included in the response
- ****exclude** (*Optional[list] = None*) – Filter which keys should be excluded from the response
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites `Client.timeout`

get_clan_war_log (*tag: clashroyale.royaleapi.utils.crtag, **params*)

Get a clan's war log

Parameters

- ***tags** (*str*) – Valid clan tags. Minimum length: 3 Valid characters: 0289PYLQGR-JCUV
- ****keys** (*Optional[list] = None*) – Filter which keys should be included in the response
- ****exclude** (*Optional[list] = None*) – Filter which keys should be excluded from the response
- ****max** (*Optional[int] = None*) – Limit the number of items returned in the response
- ****page** (*Optional[int] = None*) – Works with max, the zero-based page of the items
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites `Client.timeout`

get_constants (***params*)

Get the CR Constants

Parameters

- ****keys** (*Optional[list] = None*) – Filter which keys should be included in the response
- ****exclude** (*Optional[list] = None*) – Filter which keys should be excluded from the response
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites `Client.timeout`

get_endpoints ()

Gets a list of endpoints available in RoyaleAPI

get_full_tournaments (**params)

Get a list of tournaments that are full

****1k: Optional[int] = 0** Set to 1 to filter tournaments that have at least 1000 max players

****open: Optional[int] = 0** Set to 1 to filter tournaments that are open

****inprep: Optional[int] = 0** Set to 1 to filter tournaments that are in preperation

****joinable: Optional[int] = 0** Set to 1 to filter tournaments that are joinable

****keys: Optional[list] = None** Filter which keys should be included in the response

****exclude: Optional[list] = None** Filter which keys should be excluded from the response

****max: Optional[int] = None** Limit the number of items returned in the response

****page: Optional[int] = None** Works with max, the zero-based page of the items

****timeout: Optional[int] = None** Custom timeout that overwrites Client.timeout

get_joinable_tournaments (**params)

Get a list of tournaments that are joinable

****1k: Optional[int] = 0** Set to 1 to filter tournaments that have at least 1000 max players

****open: Optional[int] = 0** Set to 1 to filter tournaments that are open

****full: Optional[int] = 0** Set to 1 to filter tournaments that are full

****inprep: Optional[int] = 0** Set to 1 to filter tournaments that are in preperation

****keys: Optional[list] = None** Filter which keys should be included in the response

****exclude: Optional[list] = None** Filter which keys should be excluded from the response

****max: Optional[int] = None** Limit the number of items returned in the response

****page: Optional[int] = None** Works with max, the zero-based page of the items

****timeout: Optional[int] = None** Custom timeout that overwrites Client.timeout

get_known_tournaments (**params)

Get a list of queried tournaments

****keys: Optional[list] = None** Filter which keys should be included in the response

****exclude: Optional[list] = None** Filter which keys should be excluded from the response

****max: Optional[int] = None** Limit the number of items returned in the response

****page: Optional[int] = None** Works with max, the zero-based page of the items

****timeout: Optional[int] = None** Custom timeout that overwrites Client.timeout

get_open_tournaments (**params)

Get a list of open tournaments

****1k: Optional[int] = 0** Set to 1 to filter tournaments that have at least 1000 max players

****full: Optional[int] = 0** Set to 1 to filter tournaments that are full

****inprep: Optional[int] = 0** Set to 1 to filter tournaments that are in preperation

****joinable: Optional[int] = 0** Set to 1 to filter tournaments that are joinable

****keys: Optional[list] = None** Filter which keys should be included in the response

****exclude: Optional[list] = None** Filter which keys should be excluded from the response

****max: Optional[int] = None** Limit the number of items returned in the response

****page: Optional[int] = None** Works with max, the zero-based page of the items

****timeout: Optional[int] = None** Custom timeout that overwrites Client.timeout

get_player (*tags, **params)

Get a player information

Parameters

- ***tags** (*str*) – Valid player tags. Minimum length: 3 Valid characters: 0289PYLQGR-JCUV
- ****keys** (*Optional[list] = None*) – Filter which keys should be included in the response
- ****exclude** (*Optional[list] = None*) – Filter which keys should be excluded from the response
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites Client.timeout

get_player_battles (*tags, **params)

Get a player's battle log

Parameters

- ***tags** (*str*) – Valid player tags. Minimum length: 3 Valid characters: 0289PYLQGR-JCUV
- ****keys** (*Optional[list] = None*) – Filter which keys should be included in the response
- ****exclude** (*Optional[list] = None*) – Filter which keys should be excluded from the response
- ****max** (*Optional[int] = None*) – Limit the number of items returned in the response
- ****page** (*Optional[int] = None*) – Works with max, the zero-based page of the items
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites Client.timeout

get_player_chests (*tags, **params)

Get information about a player's chest cycle

Parameters

- ***tags** (*str*) – Valid player tags. Minimum length: 3 Valid characters: 0289PYLQGR-JCUV
- ****keys** (*Optional[list] = None*) – Filter which keys should be included in the response
- ****exclude** (*Optional[list] = None*) – Filter which keys should be excluded from the response
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites Client.timeout

get_player_verify (tag: *clashroyale.royaleapi.utils.crtag*, apikey: *str*, **params)

Check the API Key of a player. This endpoint has been **restricted** to certain members of the community

Parameters

- **tag** (*str*) – A valid tournament tag. Minimum length: 3 Valid characters: 0289PYLQ-GRJCUV
- **apikey** (*str*) – The API Key in the player’s settings
- ****keys** (*Optional[list] = None*) – Filter which keys should be included in the response
- ****exclude** (*Optional[list] = None*) – Filter which keys should be excluded from the response
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites Client.timeout

get_popular_clans (***params*)

Get a list of most queried clans

- **keys: Optional[list] = None** Filter which keys should be included in the response
- **exclude: Optional[list] = None** Filter which keys should be excluded from the response
- **max: Optional[int] = None** Limit the number of items returned in the response
- **page: Optional[int] = None** Works with max, the zero-based page of the items
- **timeout: Optional[int] = None** Custom timeout that overwrites Client.timeout

get_popular_decks (***params*)

Get a list of most queried decks

- **keys: Optional[list] = None** Filter which keys should be included in the response
- **exclude: Optional[list] = None** Filter which keys should be excluded from the response
- **max: Optional[int] = None** Limit the number of items returned in the response
- **page: Optional[int] = None** Works with max, the zero-based page of the items
- **timeout: Optional[int] = None** Custom timeout that overwrites Client.timeout

get_popular_players (***params*)

Get a list of most queried players

- **keys: Optional[list] = None** Filter which keys should be included in the response
- **exclude: Optional[list] = None** Filter which keys should be excluded from the response
- **max: Optional[int] = None** Limit the number of items returned in the response
- **page: Optional[int] = None** Works with max, the zero-based page of the items
- **timeout: Optional[int] = None** Custom timeout that overwrites Client.timeout

get_popular_tournaments (***params*)

Get a list of most queried tournaments

- **keys: Optional[list] = None** Filter which keys should be included in the response
- **exclude: Optional[list] = None** Filter which keys should be excluded from the response
- **max: Optional[int] = None** Limit the number of items returned in the response
- **page: Optional[int] = None** Works with max, the zero-based page of the items
- **timeout: Optional[int] = None** Custom timeout that overwrites Client.timeout

get_prep_tournaments (***params*)

Get a list of tournaments that are in preparation

****1k: Optional[int] = 0** Set to 1 to filter tournaments that have at least 1000 max players

****open: Optional[int] = 0** Set to 1 to filter tournaments that are open

****full: Optional[int] = 0** Set to 1 to filter tournaments that are full

****joinable: Optional[int] = 0** Set to 1 to filter tournaments that are joinable

****keys: Optional[list] = None** Filter which keys should be included in the response

****exclude: Optional[list] = None** Filter which keys should be excluded from the response

****max: Optional[int] = None** Limit the number of items returned in the response

****page: Optional[int] = None** Works with max, the zero-based page of the items

****timeout: Optional[int] = None** Custom timeout that overwrites Client.timeout

get_top_clans (*country_key='all', **params*)

Get a list of top clans by trophy

location_id: Optional[str] = '' A location ID or '' (global) See <https://github.com/RoyaleAPI/cr-api-data/blob/master/json/regions.json> for a list of acceptable location IDs

****keys: Optional[list] = None** Filter which keys should be included in the response

****exclude: Optional[list] = None** Filter which keys should be excluded from the response

****max: Optional[int] = None** Limit the number of items returned in the response

****page: Optional[int] = None** Works with max, the zero-based page of the items

****timeout: Optional[int] = None** Custom timeout that overwrites Client.timeout

get_top_players (*country_key='all', **params*)

Get a list of top players

location_id: Optional[str] = '' A location ID or '' (global) See <https://github.com/RoyaleAPI/cr-api-data/blob/master/json/regions.json> for a list of acceptable location IDs

****keys: Optional[list] = None** Filter which keys should be included in the response

****exclude: Optional[list] = None** Filter which keys should be excluded from the response

****max: Optional[int] = None** Limit the number of items returned in the response

****page: Optional[int] = None** Works with max, the zero-based page of the items

****timeout: Optional[int] = None** Custom timeout that overwrites Client.timeout

get_top_war_clans (*country_key='all', **params*)

Get a list of top clans by war

location_id: Optional[str] = '' A location ID or '' (global) See <https://github.com/RoyaleAPI/cr-api-data/blob/master/json/regions.json> for a list of acceptable location IDs

****keys: Optional[list] = None** Filter which keys should be included in the response

****exclude: Optional[list] = None** Filter which keys should be excluded from the response

****max: Optional[int] = None** Limit the number of items returned in the response

****page: Optional[int] = None** Works with max, the zero-based page of the items

****timeout: Optional[int] = None** Custom timeout that overwrites Client.timeout

get_tournament (*tag: clashroyale.royaleapi.utils.crtag, **params*)

Get a tournament information

Parameters

- **tag** (*str*) – A valid tournament tag. Minimum length: 3 Valid characters: 0289PYLQ-GRJCUV
- ****keys** (*Optional[list] = None*) – Filter which keys should be included in the response
- ****exclude** (*Optional[list] = None*) – Filter which keys should be excluded from the response
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites Client.timeout

get_tracking_clans (***params*)

Get a list of clans that are being tracked by having either cr-api.com or royaleapi.com in the description

Parameters

- ****keys** (*Optional[list] = None*) – Filter which keys should be included in the response
- ****exclude** (*Optional[list] = None*) – Filter which keys should be excluded from the response
- ****max** (*Optional[int] = None*) – Limit the number of items returned in the response
- ****page** (*Optional[int] = None*) – Works with max, the zero-based page of the items
- ****timeout** (*Optional[int] = None*) – Custom timeout that overwrites Client.timeout

get_version ()

Gets the version of RoyaleAPI. Returns a string

search_clans (***params*)

Search for a clan. At least one of the filters must be present

Parameters

- **name** (*Optional[str]*) – The name of a clan
- **minMembers** (*Optional[int]*) – The minimum member count of a clan
- **maxMembers** (*Optional[int]*) – The maximum member count of a clan
- **score** (*Optional[int]*) – The minimum trophy score of a clan
- ****keys** (*Optional[list] = None*) – Filter which keys should be included in the response
- ****exclude** (*Optional[list] = None*) – Filter which keys should be excluded from the response
- ****max** (*Optional[int] = None*) – Limit the number of items returned in the response
- ****page** (*Optional[int] = None*) – Works with max, the zero-based page of the items

- **timeout** (*Optional[int] = None*) – Custom timeout that overwrites Client.timeout

search_tournaments (***params*)

Search for a tournament

Parameters

- **name** (*str*) – The name of the tournament
- **keys** (*Optional[list] = None*) – Filter which keys should be included in the response
- **exclude** (*Optional[list] = None*) – Filter which keys should be excluded from the response
- **max** (*Optional[int] = None*) – Limit the number of items returned in the response
- **page** (*Optional[int] = None*) – Works with max, the zero-based page of the items
- **timeout** (*Optional[int] = None*) – Custom timeout that overwrites Client.timeout

Data Models

class clashroyale.royaleapi.models.**BaseAttrDict** (*client, data, response, cached=False, ts=None*)

This class is the base class for all models, its a wrapper around the [python-box](#) which allows access to data via dot notation, in this case, API data will be accessed using this class. This class shouldnt normally be used by the user since its a base class for the actual models returned from the client.

Example

Accessing data via dot notation:

This functionality allows this library to present API data in a clean dynamic way.

raw_data

The raw data in the form of a dictionary being used

Type dict

cached

Whether or not the data being used is cached data from the cache database.

Type bool

last_updated

When the data which is currently being used was last updated.

Type datetime.datetime

response

Response object containing headers and more information. Returns None if cached

Type requests.Response or aiohttp.ClientResponse or None

class clashroyale.royaleapi.models.**Refreshable** (*client, data, response, cached=False, ts=None*)

Mixin class for re requesting data from the api for the specific model.

refresh()

(a)sync refresh the data.

class clashroyale.royaleapi.models.**PartialTournament** (*client, data, response, cached=False, ts=None*)

class clashroyale.royaleapi.models.**PartialClan** (*client, data, response, cached=False, ts=None*)

get_clan()

(a)sync function to return clan.

class clashroyale.royaleapi.models.**PartialPlayer** (*client, data, response, cached=False, ts=None*)

get_player()

(a)sync function to return player.

class clashroyale.royaleapi.models.**PartialPlayerClan** (*client, data, response, cached=False, ts=None*)

Brief player model, does not contain full data, non refreshable.

get_clan()

(a)sync function to return clan.

get_player()

(a)sync function to return player.

class clashroyale.royaleapi.models.**Member** (*clan, data, response*)

A clan member model, keeps a reference to the clan object it came from.

get_player()

(a)sync function to return player.

class clashroyale.royaleapi.models.**FullPlayer** (*client, data, response, cached=False, ts=None*)

A clash royale player model.

get_clan()

(a)sync function to return clan.

refresh()

(a)sync refresh the data.

class clashroyale.royaleapi.models.**FullClan** (*client, data, response, cached=False, ts=None*)

A clash royale clan model, full data + refreshable.

refresh()

(a)sync refresh the data.

class clashroyale.royaleapi.models.**rlist** (*client, data, cached, ts, response*)

append()

Append object to the end of the list.

clear()

Remove all items from list.

copy()

Return a shallow copy of the list.

count()

Return number of occurrences of value.

extend()
Extend list by appending elements from the iterable.

index()
Return first index of value.
Raises `ValueError` if the value is not present.

insert()
Insert object before index.

pop()
Remove and return item at index (default last).
Raises `IndexError` if list is empty or index is out of range.

refresh()
(a)sync refresh the data.

remove()
Remove first occurrence of value.
Raises `ValueError` if the value is not present.

reverse()
Reverse *IN PLACE*.

sort()
Stable sort *IN PLACE*.

2.1.3 Exceptions

The following exceptions are thrown by the library.

exception `clashroyale.RequestError`
Base class for all errors

exception `clashroyale.StatusError` (*resp, data*)
Base class for all errors except `NotResponding` and `RatelimitDetectedError`

exception `clashroyale.NotResponding`
Raised if the API request timed out

exception `clashroyale.NetworkError`
Raised if there is an issue with the network (i.e. `aiohttp.ServerDisconnectedError` or `requests.ConnectionError`)

exception `clashroyale.BadRequest` (*resp, data*)
Raised when status code 400 is returned. Typically when at least one search parameter was not provided

exception `clashroyale.NotFoundError` (*resp, data*)
Raised if the player/clan is not found

exception `clashroyale.ServerError` (*resp, data*)
Raised if the api service is having issues

exception `clashroyale.Unauthorized` (*resp, data*)
Raised if you passed an invalid token.

exception `clashroyale.NotTrackedError` (*resp, data*)
Raised if the requested clan is not tracked (RoyaleAPI)

exception `clashroyale.RatelimitError` (*resp, data*)
Raised if ratelimit is hit

exception `clashroyale.UnexpectedError` (*resp, data*)

Raised if the error was not caught

exception `clashroyale.RatelimitErrorDetected` (*retry_when*)

Raised when a ratelimit error is detected

2.2 Setting Up Logging

clashroyale logs errors and debug information via the `logging` python module. It is strongly recommended that the logging module is configured, as no errors or warnings will be output if it is not set up. Configuration of the logging module can be as simple as

```
import logging

logging.basicConfig(level=logging.INFO)
```

Placed at the start of the application. This will output the logs from *clashroyale* as well as other libraries that uses the logging module directly to the console.

The optional `level` argument specifies what level of events to log out and can any of `CRITICAL`, `ERROR`, `WARNING`, `INFO`, and `DEBUG` and if not specified defaults to `WARNING`.

More advance setups are possible with the logging module. To for example write the logs to a file called `clashroyale.log` instead of outputting them to to the console the following snippet can be used

```
import clashroyale
import logging

logger = logging.getLogger('clashroyale')
logger.setLevel(logging.DEBUG)
handler = logging.FileHandler(filename='clashroyale.log', encoding='utf-8', mode='w')
handler.setFormatter(logging.Formatter('%(asctime)s: %(levelname)s: %(name)s:
→ %(message)s'))
logger.addHandler(handler)
```

This is recommended, especially at verbose levels such as `INFO`, and `DEBUG` as there are a lot of events logged and it would clog the stdout of your program.

Currently, the following things are logged:

- `DEBUG`: API Requests

For more information, check the documentation and tutorial of the logging module.

CHAPTER 3

Indices

- `genindex`
- `search`

A

`all_data()` (*clashroyale.official_api.models.PaginatedAttrDict method*), 11

`append()` (*clashroyale.official_api.models.rlist method*), 12

`append()` (*clashroyale.royaleapi.models.rlist method*), 22

`Async()` (*clashroyale.official_api.Client class method*), 6

`Async()` (*clashroyale.royaleapi.Client class method*), 13

B

`BadRequest`, 23

`BaseAttrDict` (*class in clashroyale.official_api.models*), 10

`BaseAttrDict` (*class in clashroyale.royaleapi.models*), 21

C

`cached` (*clashroyale.official_api.models.BaseAttrDict attribute*), 10

`cached` (*clashroyale.royaleapi.models.BaseAttrDict attribute*), 21

`clear()` (*clashroyale.official_api.models.rlist method*), 12

`clear()` (*clashroyale.royaleapi.models.rlist method*), 22

`Client` (*class in clashroyale.official_api*), 5

`Client` (*class in clashroyale.royaleapi*), 12

`copy()` (*clashroyale.official_api.models.rlist method*), 12

`copy()` (*clashroyale.royaleapi.models.rlist method*), 22

`count()` (*clashroyale.official_api.models.rlist method*), 12

`count()` (*clashroyale.royaleapi.models.rlist method*), 22

E

`extend()` (*clashroyale.official_api.models.rlist*

method), 12

`extend()` (*clashroyale.royaleapi.models.rlist method*), 23

F

`FullClan` (*class in clashroyale.official_api.models*), 12

`FullClan` (*class in clashroyale.royaleapi.models*), 22

`FullPlayer` (*class in clashroyale.official_api.models*), 11

`FullPlayer` (*class in clashroyale.royaleapi.models*), 22

G

`get_1k_tournaments()` (*clashroyale.royaleapi.Client method*), 13

`get_all_cards()` (*clashroyale.official_api.Client method*), 6

`get_all_locations()` (*clashroyale.official_api.Client method*), 6

`get_arena_image()` (*clashroyale.official_api.Client method*), 6

`get_card_info()` (*clashroyale.official_api.Client method*), 6

`get_clan()` (*clashroyale.official_api.Client method*), 6

`get_clan()` (*clashroyale.official_api.models.FullPlayer method*), 11

`get_clan()` (*clashroyale.official_api.models.PartialClan method*), 11

`get_clan()` (*clashroyale.official_api.models.PartialPlayerClan method*), 11

`get_clan()` (*clashroyale.royaleapi.Client method*), 13

`get_clan()` (*clashroyale.royaleapi.models.FullPlayer method*), 22

`get_clan()` (*clashroyale.royaleapi.models.PartialClan method*), 22

`get_clan()` (*clashroyale.royaleapi.models.PartialPlayerClan method*), 22

<code>get_clan_battles()</code> (<i>clashroyale.royaleapi.Client method</i>), 14	<code>get_player_battles()</code> (<i>clashroyale.official_api.Client method</i>), 8
<code>get_clan_history()</code> (<i>clashroyale.royaleapi.Client method</i>), 14	<code>get_player_battles()</code> (<i>clashroyale.royaleapi.Client method</i>), 17
<code>get_clan_image()</code> (<i>clashroyale.official_api.Client method</i>), 6	<code>get_player_chests()</code> (<i>clashroyale.official_api.Client method</i>), 8
<code>get_clan_members()</code> (<i>clashroyale.official_api.Client method</i>), 6	<code>get_player_chests()</code> (<i>clashroyale.royaleapi.Client method</i>), 17
<code>get_clan_tracking()</code> (<i>clashroyale.royaleapi.Client method</i>), 14	<code>get_player_verify()</code> (<i>clashroyale.official_api.Client method</i>), 8
<code>get_clan_war()</code> (<i>clashroyale.official_api.Client method</i>), 7	<code>get_player_verify()</code> (<i>clashroyale.royaleapi.Client method</i>), 17
<code>get_clan_war()</code> (<i>clashroyale.royaleapi.Client method</i>), 15	<code>get_popular_clans()</code> (<i>clashroyale.royaleapi.Client method</i>), 18
<code>get_clan_war_log()</code> (<i>clashroyale.official_api.Client method</i>), 7	<code>get_popular_decks()</code> (<i>clashroyale.royaleapi.Client method</i>), 18
<code>get_clan_war_log()</code> (<i>clashroyale.royaleapi.Client method</i>), 15	<code>get_popular_players()</code> (<i>clashroyale.royaleapi.Client method</i>), 18
<code>get_constants()</code> (<i>clashroyale.royaleapi.Client method</i>), 15	<code>get_popular_tournaments()</code> (<i>clashroyale.royaleapi.Client method</i>), 18
<code>get_datetime()</code> (<i>clashroyale.official_api.Client method</i>), 7	<code>get_prep_tournaments()</code> (<i>clashroyale.royaleapi.Client method</i>), 18
<code>get_deck_link()</code> (<i>clashroyale.official_api.Client method</i>), 7	<code>get_rarity_info()</code> (<i>clashroyale.official_api.Client method</i>), 8
<code>get_endpoints()</code> (<i>clashroyale.royaleapi.Client method</i>), 15	<code>get_top_clans()</code> (<i>clashroyale.official_api.Client method</i>), 9
<code>get_full_tournaments()</code> (<i>clashroyale.royaleapi.Client method</i>), 15	<code>get_top_clans()</code> (<i>clashroyale.royaleapi.Client method</i>), 19
<code>get_joinable_tournaments()</code> (<i>clashroyale.royaleapi.Client method</i>), 16	<code>get_top_clanwar_clans()</code> (<i>clashroyale.official_api.Client method</i>), 9
<code>get_known_tournaments()</code> (<i>clashroyale.royaleapi.Client method</i>), 16	<code>get_top_players()</code> (<i>clashroyale.official_api.Client method</i>), 9
<code>get_location()</code> (<i>clashroyale.official_api.Client method</i>), 7	<code>get_top_players()</code> (<i>clashroyale.royaleapi.Client method</i>), 19
<code>get_open_tournaments()</code> (<i>clashroyale.royaleapi.Client method</i>), 16	<code>get_top_war_clans()</code> (<i>clashroyale.royaleapi.Client method</i>), 19
<code>get_player()</code> (<i>clashroyale.official_api.Client method</i>), 8	<code>get_tournament()</code> (<i>clashroyale.official_api.Client method</i>), 9
<code>get_player()</code> (<i>clashroyale.official_api.models.Member method</i>), 11	<code>get_tournament()</code> (<i>clashroyale.royaleapi.Client method</i>), 19
<code>get_player()</code> (<i>clashroyale.official_api.models.PartialPlayer method</i>), 11	<code>get_tracking_clans()</code> (<i>clashroyale.royaleapi.Client method</i>), 20
<code>get_player()</code> (<i>clashroyale.official_api.models.PartialPlayer method</i>), 11	<code>get_version()</code> (<i>clashroyale.royaleapi.Client method</i>), 20
<code>get_player()</code> (<i>clashroyale.royaleapi.Client method</i>), 17	
<code>get_player()</code> (<i>clashroyale.royaleapi.models.Member method</i>), 22	
<code>get_player()</code> (<i>clashroyale.royaleapi.models.PartialPlayer method</i>), 22	<code>index()</code> (<i>clashroyale.official_api.models.rlist method</i>), 12
<code>get_player()</code> (<i>clashroyale.royaleapi.models.PartialPlayer method</i>), 22	<code>index()</code> (<i>clashroyale.royaleapi.models.rlist method</i>), 23

`insert()` (*clashroyale.official_api.models.rlist method*), 12
`insert()` (*clashroyale.royaleapi.models.rlist method*), 23

L

`last_updated` (*clashroyale.official_api.models.BaseAttrDict attribute*), 10
`last_updated` (*clashroyale.royaleapi.models.BaseAttrDict attribute*), 21

M

`Member` (*class in clashroyale.official_api.models*), 11
`Member` (*class in clashroyale.royaleapi.models*), 22

N

`NetworkError`, 23
`NotFoundError`, 23
`NotResponding`, 23
`NotTrackedError`, 23

P

`PaginatedAttrDict` (*class in clashroyale.official_api.models*), 11
`PartialClan` (*class in clashroyale.official_api.models*), 11
`PartialClan` (*class in clashroyale.royaleapi.models*), 22
`PartialPlayer` (*class in clashroyale.official_api.models*), 11
`PartialPlayer` (*class in clashroyale.royaleapi.models*), 22
`PartialPlayerClan` (*class in clashroyale.official_api.models*), 11
`PartialPlayerClan` (*class in clashroyale.royaleapi.models*), 22
`PartialTournament` (*class in clashroyale.royaleapi.models*), 22
`pop()` (*clashroyale.official_api.models.rlist method*), 12
`pop()` (*clashroyale.royaleapi.models.rlist method*), 23

R

`RatelimitError`, 23
`RatelimitErrorDetected`, 24
`raw_data` (*clashroyale.official_api.models.BaseAttrDict attribute*), 10
`raw_data` (*clashroyale.royaleapi.models.BaseAttrDict attribute*), 21
`refresh()` (*clashroyale.official_api.models.FullClan method*), 12
`refresh()` (*clashroyale.official_api.models.FullPlayer method*), 12
`refresh()` (*clashroyale.official_api.models.Refreshable method*), 11

`refresh()` (*clashroyale.official_api.models.rlist method*), 12
`refresh()` (*clashroyale.royaleapi.models.FullClan method*), 22
`refresh()` (*clashroyale.royaleapi.models.FullPlayer method*), 22
`refresh()` (*clashroyale.royaleapi.models.Refreshable method*), 21
`refresh()` (*clashroyale.royaleapi.models.rlist method*), 23
`Refreshable` (*class in clashroyale.official_api.models*), 10
`Refreshable` (*class in clashroyale.royaleapi.models*), 21
`remove()` (*clashroyale.official_api.models.rlist method*), 12
`remove()` (*clashroyale.royaleapi.models.rlist method*), 23
`RequestError`, 23
`response` (*clashroyale.official_api.models.BaseAttrDict attribute*), 10
`response` (*clashroyale.royaleapi.models.BaseAttrDict attribute*), 21
`reverse()` (*clashroyale.official_api.models.rlist method*), 12
`reverse()` (*clashroyale.royaleapi.models.rlist method*), 23
`rlist` (*class in clashroyale.official_api.models*), 12
`rlist` (*class in clashroyale.royaleapi.models*), 22

S

`search_clans()` (*clashroyale.official_api.Client method*), 9
`search_clans()` (*clashroyale.royaleapi.Client method*), 20
`search_tournaments()` (*clashroyale.official_api.Client method*), 10
`search_tournaments()` (*clashroyale.royaleapi.Client method*), 21
`ServerError`, 23
`sort()` (*clashroyale.official_api.models.rlist method*), 12
`sort()` (*clashroyale.royaleapi.models.rlist method*), 23
`StatusError`, 23

U

`Unauthorized`, 23
`UnexpectedError`, 23
`update_data()` (*clashroyale.official_api.models.PaginatedAttrDict method*), 11