
pychesscom

Rohan Rao

Feb 10, 2021

CONTENTS:

1	Installation	3
2	Clients	5
2.1	ChessComClient	5
2.2	BaseClient	5
3	Endpoints	7
3.1	Club	7
3.2	Country	8
3.3	Leaderboard	9
3.4	Match	10
3.5	Player	11
3.6	Puzzle	15
3.7	Streamer	15
3.8	Tournament	16
4	Utilities	19
4.1	Response	19
4.2	Route	19
5	Resources	21
	Index	23

An asynchronous Python client for Chess.com's API.

INSTALLATION

Python 3.7 or higher is required

To install stable version from PyPI (recommended):

```
$ pip install pychesscom
```

To install development version:

```
$ git clone https://github.com/vopani/pychesscom  
$ cd pychesscom  
$ python3 -m pip install -r requirements.txt
```


CLIENTS

2.1 ChessComClient

class ChessComClient (*loop=None*)

Bases: object

Class for handling Chess.com API requests.

Parameters **loop** (`asyncio.AbstractEventLoop`) – Asyncio event loop

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
```

2.2 BaseClient

class BaseClient (*loop=None*)

Bases: object

Class for handling HTTP Client requests.

Parameters **loop** (`asyncio.AbstractEventLoop`) – Asyncio event loop

async request (*route:* `pychesscom.utils.route.Route`, ***kwargs*) → `pychesscom.utils.response.Response`
HTTP request for a route.

Parameters **route** (`pychesscom.utils.route.Route`) – The route for API request

Returns Response of the API request

ENDPOINTS

3.1 Club

class Club (*client*: `pychesscom.clients.base_client.BaseClient`)

Bases: object

Class for handling endpoints of club information.

Parameters `client` (`pychesscom.clients.base_client.BaseClient`) – HTTP client for API requests

async get_details (*url_id*: *str*) → `pychesscom.utils.response.Response`

Get profile details of a club.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-club-profile>

Parameters `url_id` (*str*) – The `url_id` of a club's web page on chess.com

Returns Response of API request

Return type `pychesscom.utils.response.Response`

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.club.get_details('chess-com-developer-community')
print(response)
```

async get_matches (*url_id*: *str*) → `pychesscom.utils.response.Response`

Get team matches of a club.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-club-matches>

Parameters `url_id` (*str*) – The `url_id` of a club's web page on chess.com

Returns Response of API request

Return type `pychesscom.utils.response.Response`

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.club.get_matches('chess-com-developer-community')
print(response)
```

async get_members (*url_id: str*) → *pychesscom.utils.response.Response*

Get members of a club.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-club-members>

Parameters *url_id (str)* – The url_id of a club’s web page on chess.com

Returns Response of API request

Return type *pychesscom.utils.response.Response*

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.club.get_members('chess-com-developer-community')
print(response)
```

3.2 Country

class Country (*client: pychesscom.clients.base_client.BaseClient*)

Bases: object

Class for handling endpoints of country information.

Parameters *client (pychesscom.clients.base_client.BaseClient)* – HTTP client for API requests

async get_clubs (*iso: str*) → *pychesscom.utils.response.Response*

Get clubs of a country.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-country-clubs>

Parameters *iso (str)* – 2-character ISO 3166 code of country

Returns Response of API request

Return type *pychesscom.utils.response.Response*

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.country.get_clubs('IT')
print(response)
```

async get_details (*iso: str*) → *pychesscom.utils.response.Response*

Get profile details of a country.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-country-profile>

Parameters *iso (str)* – 2-character ISO 3166 code of country

Returns Response of API request

Return type *pychesscom.utils.response.Response*

Example:

```

from pychesscom import ChessComClient
client = ChessComClient()
response = await client.country.get_details('IT')
print(response)

```

async get_players (*iso: str*) → *pychesscom.utils.response.Response*
 Get players of a country.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-country-players>

Parameters *iso* (*str*) – 2-character ISO 3166 code of country

Returns Response of API request

Return type *pychesscom.utils.response.Response*

Example:

```

from pychesscom import ChessComClient
client = ChessComClient()
response = await client.country.get_players('IT')
print(response)

```

3.3 Leaderboard

class Leaderboard (*client: pychesscom.clients.base_client.BaseClient*)

Bases: object

Class for handling endpoints of leaderboard information.

Parameters *client* (*pychesscom.clients.base_client.BaseClient*) – HTTP client for API requests

async get_leaderboards () → *pychesscom.utils.response.Response*
 Get top-50 players of leaderboards.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-leaderboards>

Returns Response of API request

Return type *pychesscom.utils.response.Response*

Example:

```

from pychesscom import ChessComClient
client = ChessComClient()
response = await client.leaderboard.get_leaderboards()
print(response)

```

3.4 Match

class Match (*client*: `pychesscom.clients.base_client.BaseClient`)

Bases: `object`

Class for handling endpoints of team match information.

Parameters *client* (`pychesscom.clients.base_client.BaseClient`) – HTTP client for API requests

async get_board (*match_id*: `int`, *board_id*: `int`) → `pychesscom.utils.response.Response`

Get board details of a team match.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-match-board>

Parameters

- **match_id** (`int`) – The *match_id* of a team match
- **board_id** (`int`) – The *board_id* of a team match

Returns Response of API request

Return type `pychesscom.utils.response.Response`

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.match.get_board(12803, 1)
print(response)
```

async get_details (*match_id*: `int`) → `pychesscom.utils.response.Response`

Get profile details of a team match.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-match-profile>

Parameters *match_id* (`int`) – The *match_id* of a team match

Returns Response of API request

Return type `pychesscom.utils.response.Response`

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.match.get_details(12803)
print(response)
```

async get_live_board (*match_id*: `int`, *board_id*: `int`) → `pychesscom.utils.response.Response`

Get board details of a live team match.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-match-live-board>

Parameters

- **match_id** (`int`) – The *match_id* of a live team match
- **board_id** (`int`) – The *board_id* of a live team match

Returns Response of API request

Return type `pychesscom.utils.response.Response`

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.match.get_live_board(5833, 1)
print(response)
```

async get_live_details (*match_id: int*) → *pychesscom.utils.response.Response*

Get profile details of a live team match.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-match-live-profile>

Parameters *match_id* (*int*) – The *match_id* of a live team match

Returns Response of API request

Return type *pychesscom.utils.response.Response*

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.match.get_live_details(5833)
print(response)
```

3.5 Player

class Player (*client: pychesscom.clients.base_client.BaseClient*)

Bases: object

Class for handling endpoints of player information.

Parameters *client* (*pychesscom.clients.base_client.BaseClient*) – HTTP client for API requests

async get_clubs (*username: str*) → *pychesscom.utils.response.Response*

Get clubs of a player.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-player-clubs>

Parameters *username* (*str*) – The username of a player on chess.com

Returns Response of API request

Return type *pychesscom.utils.response.Response*

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.player.get_clubs('erik')
print(response)
```

async get_current_games (*username: str*) → *pychesscom.utils.response.Response*

Get current games of a player.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-games-current>

Parameters *username* (*str*) – The username of a player on chess.com

Returns Response of API request

Return type `pychesscom.utils.response.Response`

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.player.get_current_games('erik')
print(response)
```

async get_current_games_to_move (*username: str*) → `pychesscom.utils.response.Response`

Get current games of a player where it is the player's turn to move.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-games-tomove>

Parameters **username** (*str*) – The username of a player on chess.com

Returns Response of API request

Return type `pychesscom.utils.response.Response`

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.player.get_current_games_to_move('erik')
print(response)
```

async get_details (*username: str*) → `pychesscom.utils.response.Response`

Get profile details of a player.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-player>

Parameters **username** (*str*) – The username of a player on chess.com

Returns Response of API request

Return type `pychesscom.utils.response.Response`

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.player.get_details('erik')
print(response)
```

async get_games (*username: str, year: int, month: int*) → `pychesscom.utils.response.Response`

Get games of a player in a particular month.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-games-archive>

Parameters

- **username** (*str*) – The username of a player on chess.com
- **year** (*int*) – Year of archive
- **month** (*int*) – Month of archive

Returns Response of API request

Return type `pychesscom.utils.response.Response`

Example:


```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.player.get_monthly_archive('erik', 2009, 10)
print(response)
```

async get_matches (username: str) → *pychesscom.utils.response.Response*

Get team matches of a player.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-player-matches>

Parameters **username** (str) – The username of a player on chess.com

Returns Response of API request

Return type *pychesscom.utils.response.Response*

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.player.get_matches('erik')
print(response)
```

async get_monthly_archive (username: str) → *pychesscom.utils.response.Response*

Get monthly archives of a player.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-games-archive-list>

Parameters **username** (str) – The username of a player on chess.com

Returns Response of API request

Return type *pychesscom.utils.response.Response*

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.player.get_monthly_archive('erik')
print(response)
```

async get_online_status (username: str) → *pychesscom.utils.response.Response*

Get online status of a player.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-player-is-online>

Parameters **username** (str) – The username of a player on chess.com

Returns Response of API request

Return type *pychesscom.utils.response.Response*

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.player.get_online_status('erik')
print(response)
```

async get_stats (username: str) → *pychesscom.utils.response.Response*

Get stats of a player.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-player-stats>

Parameters `username` (*str*) – The username of a player on chess.com

Returns Response of API request

Return type `pychesscom.utils.response.Response`

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.player.get_stats('erik')
print(response)
```

async get_titled_players (*title: str*) → `pychesscom.utils.response.Response`

Get titled players..

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-titled>

Parameters `title` (*str*) – The title abbreviation

Returns Response of API request

Return type `pychesscom.utils.response.Response`

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.player.get_titled_players('GM')
print(response)
```

async get_tournaments (*username: str*) → `pychesscom.utils.response.Response`

Get tournaments of a player.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-player-tournaments>

Parameters `username` (*str*) – The username of a player on chess.com

Returns Response of API request

Return type `pychesscom.utils.response.Response`

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.player.get_tournaments('erik')
print(response)
```

3.6 Puzzle

class Puzzle (*client*: pychesscom.clients.base_client.BaseClient)

Bases: object

Class for handling endpoints of puzzle information.

Parameters **client** (pychesscom.clients.base_client.BaseClient) – HTTP client for API requests

async get_daily () → pychesscom.utils.response.Response

Get daily puzzle.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-daily-puzzle>

Returns Response of API request

Return type pychesscom.utils.response.Response

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.puzzle.get_daily()
print(response)
```

async get_random () → pychesscom.utils.response.Response

Get random puzzle.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-random-daily-puzzle>

Returns Response of API request

Return type pychesscom.utils.response.Response

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.puzzle.get_random()
print(response)
```

3.7 Streamer

class Streamer (*client*: pychesscom.clients.base_client.BaseClient)

Bases: object

Class for handling endpoints of streamer information.

Parameters **client** (pychesscom.clients.base_client.BaseClient) – HTTP client for API requests

async get_streamers () → pychesscom.utils.response.Response

Get streamers.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-streamers>

Returns Response of API request

Return type pychesscom.utils.response.Response

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.streamer.get_streamers()
print(response)
```

3.8 Tournament

class Tournament (*client*: pychesscom.clients.base_client.BaseClient)

Bases: object

Class for handling endpoints of tournament information.

Parameters *client* (pychesscom.clients.base_client.BaseClient) – HTTP client for API requests

async get_details (*url_id*: str) → pychesscom.utils.response.Response

Get profile details of a tournament.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-tournament-profile>

Parameters *url_id* (str) – The url_id of a tournament's web page on chess.com

Returns Response of API request

Return type pychesscom.utils.response.Response

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.tournament.get_details('-33rd-chesscom-quick-
↪knockouts-1401-1600')
print(response)
```

async get_round (*url_id*: str, *round_id*: int) → pychesscom.utils.response.Response

Get round details of a tournament.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-tournament-round>

Parameters

- *url_id* (str) – The url_id of a tournament's web page on chess.com
- *round_id* (int) – The round_id of a tournament

Returns Response of API request

Return type pychesscom.utils.response.Response

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.tournament.get_round('-33rd-chesscom-quick-knockouts-
↪1401-1600', 1)
print(response)
```

async get_round_group (*url_id: str, round_id: int, group_id: int*) → *pychesscom.utils.response.Response*

Get group details of a tournament round.

Chess.com API: <https://www.chess.com/news/view/published-data-api#pubapi-endpoint-tournament-round-group>

Parameters

- **url_id** (*str*) – The url_id of a tournament’s web page on chess.com
- **round_id** (*int*) – The round_id of a tournament
- **group_id** (*int*) – The group_id of a tournament round

Returns Response of API request

Return type *pychesscom.utils.response.Response*

Example:

```
from pychesscom import ChessComClient
client = ChessComClient()
response = await client.tournament.get_round_group('-33rd-chesscom-quick-
↳knockouts-1401-1600', 1, 1)
print(response)
```


UTILITIES

4.1 Response

```
class Response(url: str, code: int, reason: str, content_type: str, timestamp: bytes, content:
                Union[List[dict], dict])
```

Bases: object

Class for handling Chess.com API responses.

Parameters

- **url** (*str*) – URL of Chess.com API request
- **code** (*int*) – Code of API response
- **reason** (*str*) – Reason of API response
- **content_type** (*str*) – Content type of API response
- **timestamp** (*bytes*) – Timestamp of API request
- **content** (*_SpecialForm[List[dict], dict]*) – Content of API response

```
to_dict() → dict
```

Convert to dictionary.

Returns Dictionary format

Return type dict

4.2 Route

```
class Route(path: str)
```

Bases: object

Class for handling Chess.com's API routes.

Parameters **path** (*str*) – Path of route

```
BASE = 'https://api.chess.com/pub'
```

Chess.com's base API URL

RESOURCES

Introducing PyChessCom: A tutorial in Jupyter Notebook

INDEX

B

BASE (*Route attribute*), 19

BaseClient (*class in pychesscom.clients.base_client*), 5

C

ChessComClient (*class in pychesscom.clients.main_client*), 5

Club (*class in pychesscom.endpoints.club*), 7

Country (*class in pychesscom.endpoints.country*), 8

G

get_board() (*Match method*), 10

get_clubs() (*Country method*), 8

get_clubs() (*Player method*), 11

get_current_games() (*Player method*), 11

get_current_games_to_move() (*Player method*), 12

get_daily() (*Puzzle method*), 15

get_details() (*Club method*), 7

get_details() (*Country method*), 8

get_details() (*Match method*), 10

get_details() (*Player method*), 12

get_details() (*Tournament method*), 16

get_games() (*Player method*), 12

get_leaderboards() (*Leaderboard method*), 9

get_live_board() (*Match method*), 10

get_live_details() (*Match method*), 11

get_matches() (*Club method*), 7

get_matches() (*Player method*), 13

get_members() (*Club method*), 7

get_monthly_archive() (*Player method*), 13

get_online_status() (*Player method*), 13

get_players() (*Country method*), 9

get_random() (*Puzzle method*), 15

get_round() (*Tournament method*), 16

get_round_group() (*Tournament method*), 16

get_stats() (*Player method*), 13

get_streamers() (*Streamer method*), 15

get_titled_players() (*Player method*), 14

get_tournaments() (*Player method*), 14

L

Leaderboard (*class in pychesscom.endpoints.leaderboard*), 9

M

Match (*class in pychesscom.endpoints.match*), 10

P

Player (*class in pychesscom.endpoints.player*), 11

Puzzle (*class in pychesscom.endpoints.puzzle*), 15

R

request() (*BaseClient method*), 5

Response (*class in pychesscom.utils.response*), 19

Route (*class in pychesscom.utils.route*), 19

S

Streamer (*class in pychesscom.endpoints.streamer*), 15

T

to_dict() (*Response method*), 19

Tournament (*class in pychesscom.endpoints.tournament*), 16