sportmonks Documentation

Release 1.1.0

Dmitrii Izgurskii

Nov 15, 2019

CONTENTS

1 Examples 3

2 Installation 5

3 Features 7

4 *sportmonks.soccer* reference 9

Python Module Index 15

Index 17

sportmonks is a Python 3.5+ package that implements SportMonks API. While SportMonks (the company) offers data for various sports, this package implements only soccer. There are no plans to implement other sports.

Disclaimer: *sportmonks* Python package authors are not affiliated with SportMonks the company.

CHAPTER

ONE

EXAMPLES

Print today's games:

```
>>> from sportmonks.soccer import SoccerApiV2
>>> soccer = SoccerApiV2(api_token='My API token')

>>> fixtures = soccer.fixtures_today(include=('localTeam', 'visitorTeam'))
>>> for f in fixtures:
>>>     print(f['localTeam']['name'], 'plays at home against', f['visitorTeam']['name
↪'])

Randers plays at home against FC Helsingør
Celtic plays at home against Aberdeen
Hibernian plays at home against Rangers
Kilmarnock plays at home against Hearts
Silkeborg plays at home against Lyngby
Hobro plays at home against SønderjyskE
OB plays at home against AGF
AaB plays at home against København
```

CHAPTER

TWO

INSTALLATION

Latest released version can be installed with:

```
pip install sportmonks
```

Latest development version can be installed with:

```
git clone https://www.github.com/Dmitrii-I/sportmonks.git
cd sportmonks
pip install ./
```

CHAPTER

THREE

FEATURES

sportmonks features focus mainly on convenience:

- Methods return only data from the original SportMonks response, with non-data objects dropped. The result is shorter code: `[f for f in fixtures_today()]` instead of `[f for f in fixtures_today()['data']]`.
- Methods return complete data even if the underlying HTTP endpoints are paginated. Fetching all pages is arguably the most common scenario. Not having to write *for* and *while* loops to fetch additional pages results in less boilerplate code. The trade-off is that all pages are fetched even when fewer suffice.

CHAPTER

FOUR

SPORTMONKS.SOCCER REFERENCE

The *soccer* module implements the SportMonks soccer HTTP API v2.0.

class sportmonks.soccer.**SoccerApiV2**(*api_token: str*)

Bases: sportmonks._base.BaseApiV2

The SoccerApiV2 class provides SportMonks soccer API client.

Parameter api_token is the API token from the SportMonks profile web page.

http_requests_made
Number of HTTP requests made.

base_url
Base URL of the endpoint the client connects to.

api_token
API token the client uses to authenticate with SportMonks.com servers.

tz_name
Name of the timezone the returned datetimes will have.

bookmaker(*bookmaker_id: int*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return a bookmaker.

bookmakers() → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return all bookmakers.

coach(*coach_id: int*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return a coach.

commentaries(*fixture_id: int*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return commentaries of a fixture.

Not all fixtures have commentaries. If a fixture has no commentaries then an empty list is returned.

continent(*continent_id: int*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return a continent.

Parameter includes specifies objects to include in the response. Maximum level of includes allowed is 10. Valid objects are: *countries*.

continents(*includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return all continents.

Parameter includes specifies objects to include in the response. Maximum level of includes allowed is 10. Valid objects are: *countries*.

countries(*includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return all countries.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 2. Valid objects are: *continent*, *leagues*.

country(*country_id: int*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return a country.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 2. Valid objects are: *continent*, *leagues*.

fixture(*fixture_id: int*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return a fixture.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 10. Valid objects are: *localTeam*, *visitorTeam*, *substitutions*, *goals*, *cards*, *other*, *corners*, *lineup*, *bench*, *sidelined*, *stats*, *comments*, *tvstations*, *highlights*, *league*, *season*, *round*, *stage*, *referee*, *events*, *venue*, *odds*, *flatOdds*, *inplay*, *localCoach*, *visitorCoach*, *group*, *trends*.

fixture_tv_stations(*fixture_id: int*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return tv stations broadcasting specified fixture.

fixtures(*start_date: datetime.date*, *end_date: datetime.date*, *league_ids: Optional[List[int]] = None*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return fixtures between `start_date` and `end_date`. The dates are inclusive.

Parameter `league_ids` specifies leagues from which the fixtures will be returned, defaulting to all leagues.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 3. Valid objects are: *localTeam*, *visitorTeam*, *substitutions*, *goals*, *cards*, *other*, *corners*, *lineup*, *bench*, *sidelined*, *stats*, *comments*, *tvstations*, *highlights*, *league*, *season*, *round*, *stage*, *referee*, *events*, *venue*, *odds*, *flatOdds*, *inplay*, *localCoach*, *visitorCoach*, *group*, *trends*.

fixtures_in_play(*includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return in-play fixtures.

Note that in-play is defined by SportMonks as fixtures currently played, plus fixtures that will begin within 45 minutes, and fixtures that ended less than 30 minutes ago.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 10. Valid objects are: *localTeam*, *visitorTeam*, *substitutions*, *goals*, *cards*, *other*, *corners*, *lineup*, *bench*, *sidelined*, *stats*, *comments*, *tvstations*, *highlights*, *league*, *season*, *round*, *stage*, *referee*, *events*, *venue*, *odds*, *flatOdds*, *inplay*, *localCoach*, *visitorCoach*, *group*, *trends*.

fixtures_today(*league_ids: Optional[List[int]] = None*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return today's fixtures, played and to be played.

Parameter `league_ids` specifies leagues from which the fixtures will be returned, defaulting to all leagues.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 10. Valid objects are: *localTeam*, *visitorTeam*, *substitutions*, *goals*, *cards*, *other*, *corners*, *lineup*, *bench*, *sidelined*, *stats*, *comments*, *tvstations*, *highlights*, *league*, *season*, *round*, *stage*, *referee*, *events*, *venue*, *odds*, *flatOdds*, *inplay*, *localCoach*, *visitorCoach*, *group*, *trends*.

head_to_head_fixtures(*team_ids: Iterable[int], includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return all head-to-head fixtures of two teams specified by `team_ids`.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 2. Valid objects are: *localTeam*, *visitorTeam*, *substitutions*, *goals*, *cards*, *other*, *lineup*, *bench*, *stats*, *comments*, *tvstations*, *highlights*, *league*, *season*, *round*, *stage*, *referee*, *events*, *venue*, *trends*.

in_play_odds(*fixture_id: int*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return in-play odds of a fixture.

league(*league_id: int*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return a league.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 2. Valid objects are: *country*, *season*, *seasons*. The *season* include Return current season of the league. The *seasons* include Return all seasons of the league, including the current season.

leagues(*includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return all leagues.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 2. Valid objects are: *country*, *season*, *seasons*. The *season* include Return current season of the league. The *seasons* include Return all seasons of the league, including the current season.

market(*market_id: int*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return a market.

markets() → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return all betting markets, e.g. '3Way Result', 'Home/Away'.

meta() → Dict[str, Any]
Return meta data that includes your SportMonks plan, subscription, and available sports.

player(*player_id: int*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return a player.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 2. Some of the valid objects are: *position*, *team*, *stats*, *trophies*, *sidelined*, *transfers*.

pre_match_odds(*fixture_id: int*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return pre-match odds of a fixture.

round(*round_id: int*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return a round.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 2. Valid objects are: *fixtures*, *results*, *season*, *league*.

rounds(*season_id: int*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return rounds of a season.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 2. Valid objects are: *fixtures*, *results*, *season*, *league*.

season(*season_id: int*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return a season.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 10. Valid objects are: *league*, *stages*, *rounds*, *fixtures*, *upcoming*, *results*, *groups*.

season_results(*season_id: int*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return completed fixtures of a season.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 1. Valid objects are: *localTeam*, *visitorTeam*, *substitutions*, *goals*, *cards*, *other*, *corners*, *lineup*, *bench*, *sidelined*, *stats*, *comments*, *tvstations*, *highlights*, *league*, *season*, *round*, *stage*, *referee*, *events*, *venue*, *odds*, *flatOdds*, *inplay*, *localCoach*, *visitorCoach*, *group*, *trends*.

season_stages(*season_id: int*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return stages of a season.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 2. Valid objects are: *fixtures*, *results*, *season*, *league*.

season_venues(*season_id: int*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return venues of specified season.

seasons(*includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return all seasons.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 2. Valid objects are: *league*, *stages*, *rounds*, *fixtures*, *upcoming*, *results*, *groups*.

squad(*season_id: int*, *team_id: int*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return a squad. A squad is a set of players that played for a team during a season.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 3. Valid objects are: *player*, *position*.

stage(*stage_id: int*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return a stage.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 2. Valid objects are: *fixtures*, *results*, *season*, *league*.

standings(*season_id: int*, *live: bool = False*, *group_id: Optional[int] = None*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return standings of a season.

Parameter `live` specifies whether standings should also take into account in-play fixtures.

Parameter `group_id` specifies for which group to return the standings. Groups are found in tournaments like the FIFA World Cup (e.g. Group A has group ID 185).

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 2. Valid objects are: *team*, *league*, *season*, *round*, *stage*.

team(*team_id: int*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return a team.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 3. Valid objects are: *country*, *squad*, *coach*, *transfers*, *sidelined*, *stats*, *venue*, *fifaranking*, *uefaranking*, *visitorFixtures*, *localFixtures*, *visitorResults*, *localResults*, *latest*, *upcoming*.

team_fixtures(*start_date: datetime.date*, *end_date: datetime.date*, *team_id: int*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return fixtures between `start_date` and `end_date` for a team specified by `team_id`.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 3. Valid objects are: *localTeam*, *visitorTeam*, *substitutions*, *goals*, *cards*, *other*, *corners*, *lineup*, *bench*, *sidelined*, *stats*, *comments*, *tvstations*, *highlights*, *league*, *season*, *round*, *stage*, *referee*, *events*, *venue*, *odds*, *flatOdds*, *inplay*, *localCoach*, *visitorCoach*, *group*, *trends*.

team_stats(*team_id: int*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return stats of a team.

teams(*season_id: int*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return all teams that played during a season.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 3. Valid objects are: *country*, *squad*, *coach*, *transfers*, *sidelined*, *stats*, *venue*, *fifaranking*, *uefaranking*, *visitorFixtures*, *localFixtures*, *visitorResults*, *localResults*, *latest*, *upcoming*.

top_scorers(*season_id: int*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return top scorers of a season.

Three types of top scorers are returned: most goals, most assists, and most cards.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 3. Valid objects are: *goalscorers.player*, *goalscorers.team cardscorers.player*, *cardscorers.team*, *assistscorers.player*, *assistscorers.team*.

venue(*venue_id: int*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return a venue.

video_highlights(*fixture_id: Optional[int] = None*, *includes: Optional[Iterable[str]] = None*) → Union[Dict[Any, Any], List[Dict[Any, Any]]]
Return links to video highlights of all fixtures or of a fixture specified by `fixture_id`.

Parameter `includes` specifies objects to include in the response. Maximum level of includes allowed is 2. Valid objects are: *fixture*.

PYTHON MODULE INDEX

S

sportmonks.soccer, 9

INDEX

A

api_token (*sportmonks.soccer.SoccerApiV2 attribute*), 9

B

base_url (*sportmonks.soccer.SoccerApiV2 attribute*), 9
bookmaker() (*sportmonks.soccer.SoccerApiV2 method*), 9
bookmakers() (*sportmonks.soccer.SoccerApiV2 method*), 9

C

coach() (*sportmonks.soccer.SoccerApiV2 method*), 9
commentaries() (*sportmonks.soccer.SoccerApiV2 method*), 9
continent() (*sportmonks.soccer.SoccerApiV2 method*), 9
continents() (*sportmonks.soccer.SoccerApiV2 method*), 9
countries() (*sportmonks.soccer.SoccerApiV2 method*), 9
country() (*sportmonks.soccer.SoccerApiV2 method*), 10

F

fixture() (*sportmonks.soccer.SoccerApiV2 method*), 10
fixture_tv_stations() (*sportmonks.soccer.SoccerApiV2 method*), 10
fixtures() (*sportmonks.soccer.SoccerApiV2 method*), 10
fixtures_in_play() (*sportmonks.soccer.SoccerApiV2 method*), 10
fixtures_today() (*sportmonks.soccer.SoccerApiV2 method*), 10

H

head_to_head_fixtures() (*sportmonks.soccer.SoccerApiV2 method*), 10
http_requests_made (*sportmonks.soccer.SoccerApiV2 attribute*), 9

I

in_play_odds() (*sportmonks.soccer.SoccerApiV2 method*), 11

L

league() (*sportmonks.soccer.SoccerApiV2 method*), 11
leagues() (*sportmonks.soccer.SoccerApiV2 method*), 11

M

market() (*sportmonks.soccer.SoccerApiV2 method*), 11
markets() (*sportmonks.soccer.SoccerApiV2 method*), 11
meta() (*sportmonks.soccer.SoccerApiV2 method*), 11

P

player() (*sportmonks.soccer.SoccerApiV2 method*), 11
pre_match_odds() (*sportmonks.soccer.SoccerApiV2 method*), 11

R

round() (*sportmonks.soccer.SoccerApiV2 method*), 11
rounds() (*sportmonks.soccer.SoccerApiV2 method*), 11

S

season() (*sportmonks.soccer.SoccerApiV2 method*), 11
season_results() (*sportmonks.soccer.SoccerApiV2 method*), 12
season_stages() (*sportmonks.soccer.SoccerApiV2 method*), 12
season_venues() (*sportmonks.soccer.SoccerApiV2 method*), 12
seasons() (*sportmonks.soccer.SoccerApiV2 method*), 12
SoccerApiV2 (*class in sportmonks.soccer*), 9
sportmonks.soccer (*module*), 9
squad() (*sportmonks.soccer.SoccerApiV2 method*), 12

stage() (*sportmonks.soccer.SoccerApiV2 method*), 12
standings() (*sportmonks.soccer.SoccerApiV2 method*), 12

T

team() (*sportmonks.soccer.SoccerApiV2 method*), 12
team_fixtures() (*sportmonks.soccer.SoccerApiV2 method*), 12
team_stats() (*sportmonks.soccer.SoccerApiV2 method*), 13
teams() (*sportmonks.soccer.SoccerApiV2 method*), 13
top_scorers() (*sportmonks.soccer.SoccerApiV2 method*), 13
tz_name (*sportmonks.soccer.SoccerApiV2 attribute*), 9

V

venue() (*sportmonks.soccer.SoccerApiV2 method*), 13
video_highlights() (*sportmonks.soccer.SoccerApiV2 method*), 13