

Specatate888 API Documentation

Overview:

888sports is a sports betting platform. The main functional areas are:

- Manage data about sporting events to allow users to place bets.
- Provide API to receive data from external providers and update our system with the latest data about events in real-time.
- Provide access to the support team to allow them to see the most recent data for each event and to query data.

The API implementation serves the below details:

Data:

- Markets are unique per Sport
- Selections are unique per Market

Receiving data:

- For our purposes, we can assume this API will be used by a single provider, so no need to keep track of which provider is sending the messages.

Receiving request from the support team.

The implementation is in python Django. Django class models are created to hold the data and has been populated into the development sqlite3 data base. The data is retrieved from here and events of updating and adding new entry are done into the same.

Postman application has been used to check the functionalities and working of the API's.

A requirement.txt and a pip install will set up the dependencies.

The API can be verified by using postman application or the Django rest framework web UI.

Logs are also captured in the logs directory.

For all the API's with ordering field, a value with "-" if passed then the ordering is reversed (descending order), by default the value is ascending(i.e., without the "-" value)

Attached below are the API functionalities.

- List all matches, general API endpoint
- <http://127.0.0.1:8000/api/match/>

Postman

File Edit View Help

+ New Import Runner My Workspace Invite

Filter

History Collections APIs

Save Responses Clear all

Today

GET 127.0.0.1:8000/api/match/

GET spectate888

No Environment

Examples 0 BUILD

Send Save

Params Authorization Headers (6) Body Pre-request Script Tests Settings

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (9) Test Results

Status: 200 OK Time: 10 ms Size: 1.03 KB Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   {
3     "id": 2121032861909885786,
4     "url": "http://127.0.0.1:8000/api/match/2121032861909885786",
5     "name": "Eiber vs Espanyol",
6     "start_time": "2017-01-05T09:30:00Z"
7   },
8   {
9     "id": 3211032861909885564,
10    "url": "http://127.0.0.1:8000/api/match/3211032861909885564",
11    "name": "Manchester United vs Chelsea",
12    "start_time": "2020-03-15T06:30:00Z"
13  },
14  {
15    "id": 3211032861909885654,
16    "url": "http://127.0.0.1:8000/api/match/3211032861909885654",
17    "name": "Manchester City vs Arsenal",
18    "start_time": "2020-09-10T01:30:00Z"
19  },
20  }

```

Find and Replace Console

Bootcamp Build Browse

- Get match by name
- [http:// 127.0.0.1:8000/api/match/?name=Real Madrid vs Barcelona](http://127.0.0.1:8000/api/match/?name=Real Madrid vs Barcelona)

spectate888 Examples 0 BUILD Send Save

GET 127.0.0.1:8000/api/match/?name=Real Madrid vs Barcelona

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies Code

Query Params

	KEY	VALUE	DESCRIPTION	***	Bulk Edit
<input checked="" type="checkbox"/>	name	Real Madrid vs Barcelona			
	Key	Value	Description		

Body Cookies Headers (9) Test Results Status: 200 OK Time: 11 ms Size: 434 B Save Response

Pretty Raw Preview Visualize JSON Copy Search

```

1 {
2   "id": 8661032861909884224,
3   "url": "http://127.0.0.1:8000/api/match/8661032861909884224",
4   "name": "Real Madrid vs Barcelona",
5   "start_time": "2018-05-21T05:30:00Z"
6 }
7
8

```

- List a game's details by its id
- <http://127.0.0.1:8000/api/match/8661032861909884224>

The screenshot shows a REST client interface with a GET request to `127.0.0.1:8000/api/match/8661032861909884224`. The response is a JSON object with the following structure:

```
1 {
2   "id": 8661032861909884224,
3   "url": "http://127.0.0.1:8000/api/match/8661032861909884224",
4   "name": "Real Madrid vs Barcelona",
5   "startTime": "05/21/2018, 05:30:00",
6   "sport": {
7     "id": 221,
8     "name": "Football"
9   },
10  "markets": [
11    {
12      "id": 385086549360973392,
13      "selections": [
14        {
15          "id": 573766688266680774,
16          "name": "Barcelona",
17          "odds": 1.01
18        },
19        {
20          "id": 8243901714083343527,
21          "name": "Real Madrid",
22          "odds": 1.01
23        }
24      ],
25      "name": "Winner"
```

- Retrieve matches ordered by start time and sport.
- [http:// 127.0.0.1:8000/api/match/?sport=football&ordering=start_time](http://127.0.0.1:8000/api/match/?sport=football&ordering=start_time)

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** 127.0.0.1:8000/api/match/?sport=football&ordering=start_time
- Status:** 200 OK
- Time:** 14 ms
- Size:** 1.03 KB
- Response Type:** JSON

The JSON response is displayed in the 'Body' tab, showing a list of football matches ordered by start time:

```
{
  "id": 3451032861909885675,
  "url": "http://127.0.0.1:8000/api/match/3451032861909885675",
  "name": "Sevilla vs Valencia",
  "start_time": "2015-10-10T12:30:00Z"
},
{
  "id": 2121032861909885786,
  "url": "http://127.0.0.1:8000/api/match/2121032861909885786",
  "name": "Eiber vs Espanyol",
  "start_time": "2017-01-05T09:30:00Z"
},
{
  "id": 8661032861909884224,
  "url": "http://127.0.0.1:8000/api/match/8661032861909884224",
  "name": "Real Madrid vs Barcelona",
  "start_time": "2018-05-21T05:30:00Z"
},
{
  "id": 3211032861909885564,
  "url": "http://127.0.0.1:8000/api/match/3211032861909885564",
  "name": "Manchester United vs Chelsea",
  "start_time": "2020-03-15T06:30:00Z"
},
{
  "id": 3211032861909885654,
  "url": "http://127.0.0.1:8000/api/match/3211032861909885654",
  "name": "Manchester City vs Arsenal",
  "start_time": "2020-09-10T01:30:00Z"
}
```

- Retrieve matches by sport type
- [http:// 127.0.0.1:8000/api/match/?sport=football](http://127.0.0.1:8000/api/match/?sport=football)

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** 127.0.0.1:8000/api/match/?sport=football
- Status:** 200 OK
- Time:** 12 ms
- Size:** 1.03 KB
- Response Body (JSON):**

```
{
  "id": 2121032861909885786,
  "url": "http://127.0.0.1:8000/api/match/2121032861909885786",
  "name": "Eiber vs Espanyol",
  "start_time": "2017-01-05T09:30:00Z"
},
{
  "id": 3211032861909885564,
  "url": "http://127.0.0.1:8000/api/match/3211032861909885564",
  "name": "Manchester United vs Chelsea",
  "start_time": "2020-03-15T06:30:00Z"
},
{
  "id": 3211032861909885654,
  "url": "http://127.0.0.1:8000/api/match/3211032861909885654",
  "name": "Manchester City vs Arsenal",
  "start_time": "2020-09-10T01:30:00Z"
},
{
  "id": 3451032861909885675,
  "url": "http://127.0.0.1:8000/api/match/3451032861909885675",
  "name": "Sevilla vs Valencia",
  "start_time": "2015-10-10T12:30:00Z"
},
{
  "id": 8661032861909884224,
  "url": "http://127.0.0.1:8000/api/match/8661032861909884224",
  "name": "Real Madrid vs Barcelona",
}
```

- Retrieve matches by ordering by name
- [http:// 127.0.0.1:8000/api/match/?ordering=name](http://127.0.0.1:8000/api/match/?ordering=name)

GET 127.0.0.1:8000/api/match/?ordering=name Send Save

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies Code

Body Cookies Headers (9) Test Results Status: 200 OK Time: 10 ms Size: 1.03 KB Save Response

Pretty Raw Preview Visualize JSON

```
3 {
4   "id": 2121032861909885786,
5   "url": "http://127.0.0.1:8000/api/match/2121032861909885786",
6   "name": "Eibar vs Espanyol",
7   "start_time": "2017-01-05T09:30:00Z"
8 },
9 {
10  "id": 3211032861909885654,
11  "url": "http://127.0.0.1:8000/api/match/3211032861909885654",
12  "name": "Manchester City vs Arsenal",
13  "start_time": "2020-09-10T01:30:00Z"
14 },
15 {
16  "id": 321103286190988564,
17  "url": "http://127.0.0.1:8000/api/match/321103286190988564",
18  "name": "Manchester United vs Chelsea",
19  "start_time": "2020-03-15T06:30:00Z"
20 },
21 {
22  "id": 8661032861909884224,
23  "url": "http://127.0.0.1:8000/api/match/8661032861909884224",
24  "name": "Real Madrid vs Barcelona",
25  "start_time": "2018-05-21T05:30:00Z"
26 },
27 {
28  "id": 3451032861909885675,
29  "url": "http://127.0.0.1:8000/api/match/3451032861909885675",
30  "name": "Sevilla vs Valencia",
31  "start_time": "2015-10-10T12:30:00Z"
32 }
```

Bootcamp Build Browse

- Retrieve matches by ordering by start_time
- `http://127.0.0.1:8000/api/match/?ordering=start_time`

The screenshot shows a REST client interface with a GET request to `127.0.0.1:8000/api/match/?ordering=start_time`. The response is a JSON array of 5 match objects, ordered by start_time. The status is 200 OK, time is 8 ms, and size is 1.03 KB.

```
3 {
4   "id": 3451032861909885675,
5   "url": "http://127.0.0.1:8000/api/match/3451032861909885675",
6   "name": "Sevilla vs Valencia",
7   "start_time": "2015-10-10T12:30:00Z"
8 },
9 {
10  "id": 2121032861909885786,
11  "url": "http://127.0.0.1:8000/api/match/2121032861909885786",
12  "name": "Eiber vs Espanyol",
13  "start_time": "2017-01-05T09:30:00Z"
14 },
15 {
16  "id": 8661032861909884224,
17  "url": "http://127.0.0.1:8000/api/match/8661032861909884224",
18  "name": "Real Madrid vs Barcelona",
19  "start_time": "2018-05-21T05:30:00Z"
20 },
21 {
22  "id": 3211032861909885564,
23  "url": "http://127.0.0.1:8000/api/match/3211032861909885564",
24  "name": "Manchester United vs Chelsea",
25  "start_time": "2020-03-15T06:30:00Z"
26 },
27 {
28  "id": 3211032861909885654,
29  "url": "http://127.0.0.1:8000/api/match/3211032861909885654",
30  "name": "Manchester City vs Arsenal",
31  "start_time": "2020-09-10T01:30:00Z"
32 }
```


- Retrieve matches by ordering by sports name - Golf
- [http:// 127.0.0.1:8000/api/match/?sport=golf](http://127.0.0.1:8000/api/match/?sport=golf)

The screenshot displays a REST client interface with the following details:

- Request:** Method: GET, URL: `127.0.0.1:8000/api/match/?sport=golf`. Buttons for "Send" and "Save" are visible.
- Response:** Status: 200 OK, Time: 8 ms, Size: 421 B. A "Save Response" button is present.
- Body:** The response is displayed in JSON format (Pretty view). The JSON object contains:

```
{  "id": "1221032861909885655",  "url": "http://127.0.0.1:8000/api/match/1221032861909885655",  "name": "random name",  "start_time": "2016-07-28T03:30:00Z"}
```
- Footer:** Includes a "Bootcamp" logo, a "Build" button, a "Browse" button, and several utility icons.

- Retrieve matches by ordering by sports – Golf ID (more than 2 selections case)
- <http://127.0.0.1:8000/api/match/1221032861909885655>

The screenshot shows a REST client interface with a GET request to `127.0.0.1:8000/api/match/1221032861909885655`. The response is a JSON object with the following structure:

```
2  {
3    "id": 1221032861909885655,
4    "url": "http://127.0.0.1:8000/api/match/1221032861909885655",
5    "name": "random name",
6    "startTime": "07/28/2016, 03:30:00",
7    "sport": {
8      "id": 225,
9      "name": "Golf"
10   },
11   "markets": [
12     {
13       "id": 100,
14       "selections": [
15         {
16           "id": 10,
17           "name": "John",
18           "odds": 2.01
19         },
20         {
21           "id": 11,
22           "name": "Jack",
23           "odds": 4.5
24         },
25         {
26           "id": 12,
27           "name": "Kevin",
28           "odds": 1.9
29         }
30       ]
31     }
32   ]
33 }
```

The interface includes tabs for Params, Authorization, Headers (6), Body, Pre-request Script, Tests, and Settings. The Body tab is active, showing the JSON response. The status bar indicates a 200 OK response with a 10 ms timer and 612 B size. The bottom of the interface shows a Bootcamp logo and buttons for Build and Browse.