

Betting API by Salman

This project provides a RESTful sports betting API, offering information about in-season sports events and their respective odds/rates. The API utilizes Python Flask and external APIs to handle HTTP requests and responses.

Behind the scenes, the API gets data from the-odds-api.com and betsapi.com and provides a blended version of API from both providers

Features

- Retrieve information about active sports events
 - Fetch odds/rates for specific sports events
 - Get event scores
 - Retrieve upcoming events
 - Retrieve single match details
-

Note

API URL Root is <https://salman138.pythonanywhere.com>

API Endpoints

1. Get Active Events

- **URL:** `/events/<string:sport_name>`
- **Method:** `GET`
- **Description:** Retrieve active events / leagues for a specific sport.
- **Parameters:**
 - `sport_name` (string): Name of the sport. Use 'all' to retrieve events for all sports.
- **Response:**
 - Success: Returns a JSON object containing active events.
 - Error: Returns an error message if the sport name is invalid.

Usage Example

Request

```
GET /events/Cricket
```

Response

```
{
  "response": [
    {
      "active events": [
        {
          "active events": [
            {
              "away_team": "Punjab Kings",
              "commence_time": "2024-03-30T14:00:00Z",
              "home_team": "Lucknow Super Giants",
              "id": "c0fb20cddc2f8c02abca5c8c9b3c24d6",
              "sport_key": "cricket_ipl",
              "sport_title": "IPL"
            },
            {
              "away_team": "Sunrisers Hyderabad",
              "commence_time": "2024-03-31T10:00:00Z",
              "home_team": "Gujarat Titans",
              "id": "018ff754b6dd6ce468c2bc92c7352d18",
              "sport_key": "cricket_ipl",
              "sport_title": "IPL"
            },
            {
              "away_team": "Chennai Super Kings",
              "commence_time": "2024-03-31T14:00:00Z",
              "home_team": "Delhi Capitals",
              "id": "90105b4f814cb9158e133b4df14edc0b",
              "sport_key": "cricket_ipl",
              "sport_title": "IPL"
            },
            {
              "away_team": "Rajasthan Royals",
              "commence_time": "2024-04-01T14:00:00Z",
              "home_team": "Mumbai Indians",
              "id": "08a240024a4c6e976f44574309347428",
              "sport_key": "cricket_ipl",
              "sport_title": "IPL"
            }
          ],
          "description": "Indian Premier League",
          "event title": "IPL"
        },
        {
          "active events": [
            {
              "away_team": "Sri Lanka",
              "commence_time": "2024-03-31T03:00:00Z",
              "home_team": "Bangladesh",
              "id": "a485a03752de41b94c5ffbc76f9a45da",
              "sport_key": "cricket_test_match",
              "sport_title": "Test Matches"
            }
          ]
        }
      ],
    }
  ],
}
```

```
        "description": "International Test Matches",
        "event title": "Test Matches"
    }
],
"source": "OddsAPI",
"sport name": "Cricket"
},
{
    "active events": [
        {
            "away": {
                "id": "10526903",
                "name": "Punjab Kings"
            },
            "commence_time": "2024-03-30T14:00:00Z",
            "event title": "Indian Premier League",
            "event_key": "153046408",
            "home": {
                "id": "10824353",
                "name": "Lucknow Super Giants"
            }
        }
    ],
    "source": "BetsAPI",
    "sport name": "Cricket",
    "success": true
}
],
"success": true
}
```

2. Get Event Odds

- **URL:** `/odds/<string:sport_name>/<string:event_key>`
- **Method:** `GET`
- **Description:** Retrieve data (such as odds/rates) for a specific event.
- **Parameters:**
 - `sport_name` (string): Name of the sport.
 - `event_key` (string): Unique identifier for the event.
- **Response:**
 - Success: Returns a JSON object containing event data.
 - Error: Returns an error message if the event key is invalid.

Usage Example

Case 1: If the event source is Odds API, the event key will be alphabetical like `cricket_psl` or `cricket_ipl` etc.

Request

```
GET /odds/Cricket/cricket_ip1
```

Response

Response too long to include in Readme file

Go to this link to view response

https://salman138.pythonanywhere.com/odds/Cricket/cricket_ip1

Case 2: If the event source is Bets API, the event key will be numerical like **153046408** etc.

Request

```
GET /odds/Cricket/153046408
```

Response

Response too long to include in Readme file

Go to this link to view response:

<https://salman138.pythonanywhere.com/odds/Cricket/153046408>

3. Get Event Scores

- **URL:** `/scores/<string:sport_name>/<string:event_key>`
- **Method:** `GET`
- **Description:** Retrieve scores for a specific event.
- **Parameters:**
 - `sport_name` (string): Name of the sport.
 - `event_key` (string): Unique identifier for the event.
- **Response:**
 - Success: Returns a JSON object containing event scores.
 - Error: Returns an error message if the event key is invalid or scores are not available.

Usage Example

Case 1: If the event source is Odds API, the event key will be alphabetical like **cricket_test_match** or **cricket_psl** etc.

Request

```
GET /scores/Cricket/cricket_test_match
```

Response

Response too long to include in Readme file

Go to this link to view response

https://salman138.pythonanywhere.com/scores/Cricket/cricket_test_match

Case 2: If the event source is Bets API, the event key will be numerical like 153148333 etc.

Request

```
GET /scores/Cricket/153148333
```

Response

response too long to include in Readme file

Go to this link to view response

<https://salman138.pythonanywhere.com/scores/Cricket/153148333>

4. Get single match

- **URL:** `/match/<string:sport_name>/<string:event_key>`
- **Method:** GET
- **Description:** Retrieve scores for a specific event.
- **Parameters:**
 - `sport_name` (string): Name of the sport.
 - `event_key` (string): Unique identifier for the event.
 - `match_id` (string): Url encoded parameter required only in case when the source of the event / match is Odds API. You can find **match_id** in the response from `/events` endpoint discussed above, with a name not **match_id** but with a different name **id**. Get it from there and pass in the URL such as `?match_id=<match id here>`
- **Response:**
 - Success: Returns a JSON object containing event scores.
 - Error: Returns an error message if the event key is invalid or scores are not available.

Usage Example

Case 1: If the event source is Odds API, the event key will be alphabetical like `cricket_test_match` or `cricket_psl` etc. You have to pass `match_id` also as a URL encoded parameter as mentioned above

Where to get match_id:

Before you can get a single match whose source is Odds API, you should have match_id. First you have to hit the `/events` endpoint. Then In the response you can get `id`. You will pass this `id` as `?match_id=` in `/match` endpoint.

For example this is a part of response from `/events/Cricket`. You have to get `"id": "a485a03752de41b94c5ffbc76f9a45da"` from here and pass it in the `match` endpoint as `?match_id=a485a03752de41b94c5ffbc76f9a45da`

```
{
  "response" : [
    {
      "active events" : [
        {
          "active events": [
            {
              "away_team": "Sri Lanka",
              "commence_time": "2024-03-31T04:00:00Z",
              "home_team": "Bangladesh",
              "id": "a485a03752de41b94c5ffbc76f9a45da",
              "sport_key": "cricket_test_match",
              "sport_title": "Test Matches"
            }
          ],
          "description": "International Test Matches",
          "event title": "Test Matches"
        }
      ]
    }
  ]
  "source": "OddsAPI",
  "sport name": "Cricket"
}
```

Request

```
GET /match/Cricket/cricket_test_match?match_id=a485a03752de41b94c5ffbc76f9a45da
```

Response

```
{
  "response": {
    "match": {
      "away_team": "Sri Lanka",
      "commence_time": "2024-03-31T04:00:00Z",
      "home_team": "Bangladesh",
      "id": "a485a03752de41b94c5ffbc76f9a45da",
      "sport_key": "cricket_test_match",
      "sport_title": "Test Matches"
    },
    "source": "OddsAPI",
    "sport name": "Cricket"
  },
  "success": true
}
```

Case 2: If the event source is Bets API, the event key will be numerical like 153148333 etc. Now this time you do not need to give `match_id` because the source now is Bets API. You just need to give sport name and `event_key` in the request URL

Request

```
GET /match/Cricket/153088197
```

Response

```
{
  "response": {
    "match": {
      "away": {
        "id": "10416949",
        "name": "Sri Lanka"
      },
      "commence_time": "2024-03-30T05:00:00Z",
      "event title": "Bangladesh vs Sri Lanka - 2nd Test",
      "event_key": "153088197",
      "home": {
        "id": "10426216",
        "name": "Bangladesh"
      }
    },
    "source": "BetsAPI",
    "sport name": "Cricket"
  },
  "success": true
}
```

5. Get Upcoming Events

- **URL:** `/upcoming/<string:sport_name>`
- **Method:** `GET`
- **Description:** Retrieve upcoming events for a specific sport.
- **Parameters:**
 - `sport_name` (string): Name of the sport. Use 'all' to retrieve upcoming events for all sports.
- **Response:**
 - Success: Returns a JSON object containing upcoming events.
 - Error: Returns an error message if the sport name is invalid.

Usage Example

Request

```
GET /upcomming/Cricket
```

Response

```
Response too long to include in Readme file
```

View response by going to this link

<https://salman138.pythonanywhere.com/upcomming/Cricket>

License

This project is licensed under the MIT License - see the [LICENSE](#) file for details.

This README.md provides an overview of the API, including setup instructions, available endpoints, usage examples, contribution guidelines, and licensing information.

[Click here to go to the Github Repository](#)

If you have any questions or encounter any issues, feel free to contact [Salman Khokhar](#).