API-FOOTBALL (3.9.3)

support: https://dashboard.api-football.com | URL: https://www.api-football.com

Introduction

Welcome to Api-Football! You can use our API to access all API endpoints, which can get information about Football Leagues & Cups.

We have language bindings in C, C#, cURL, Dart, Go, Java, Javascript, NodeJs, Objective-c, OCaml, Php, PowerShell, Python, Ruby, Shell and Swift! You can view code examples in the dark area to the right, and you can switch the programming language of the examples with the tabs in the top right.

The update frequency indicated in the documentation is given as an indication and may vary for certain competitions.

Authentication

We uses API keys to allow access to the API. You can register a new API key in our dashboard.

API-SPORTS : https://v3.football.api-sports.io/

Our API expects for the API key to be included in all API requests to the server in a header that looks like the following:

> Make sure to replace `XxXxXxXxXxXxXxXxXxXxXxXxXx` with your API key.

REQUESTS HEADERS & CORS

The API is configured to work only with **GET** requests and allows only the headers listed below:

- `x-apisports-key`

If you make non-GET requests or add headers that are not in the list, you will receive an error from the API.

Some frameworks *(especially in JS, nodeJS..)* automatically add extra headers, you have to make sure to remove them in order to get a response from the API.

API-SPORTS Account

If you decided to subscribe directly on our site, you have a dashboard at your disposal at the following url: dashboard

It allows you to:

- To follow your consumption in real time
- Manage your subscription and change it if necessary
- Check the status of our servers
- Test all endpoints without writing a line of code.

You can also consult all this information directly through the API by calling the endpoint `status`.

> This call does not count against the daily quota.

```
get("https://v3.football.api-sports.io/status");

// response
{
    "get": "status",
    "parameters": [],
    "errors": [],
    "results": 1,
    "response": {
        "account": {
            "firstname": "xxxx",
            "lastname": "XXXXXX",
            "email": "xxx@xxx.com"
        },
        "subscription": {
            "plan": "Free",
            "end": "2020-04-10T23:24:27+00:00",
            "active": true
        },
        "requests": {
            "current": 12,
            "limit_day": 100
        }
    }
}
```

Headers sent as response

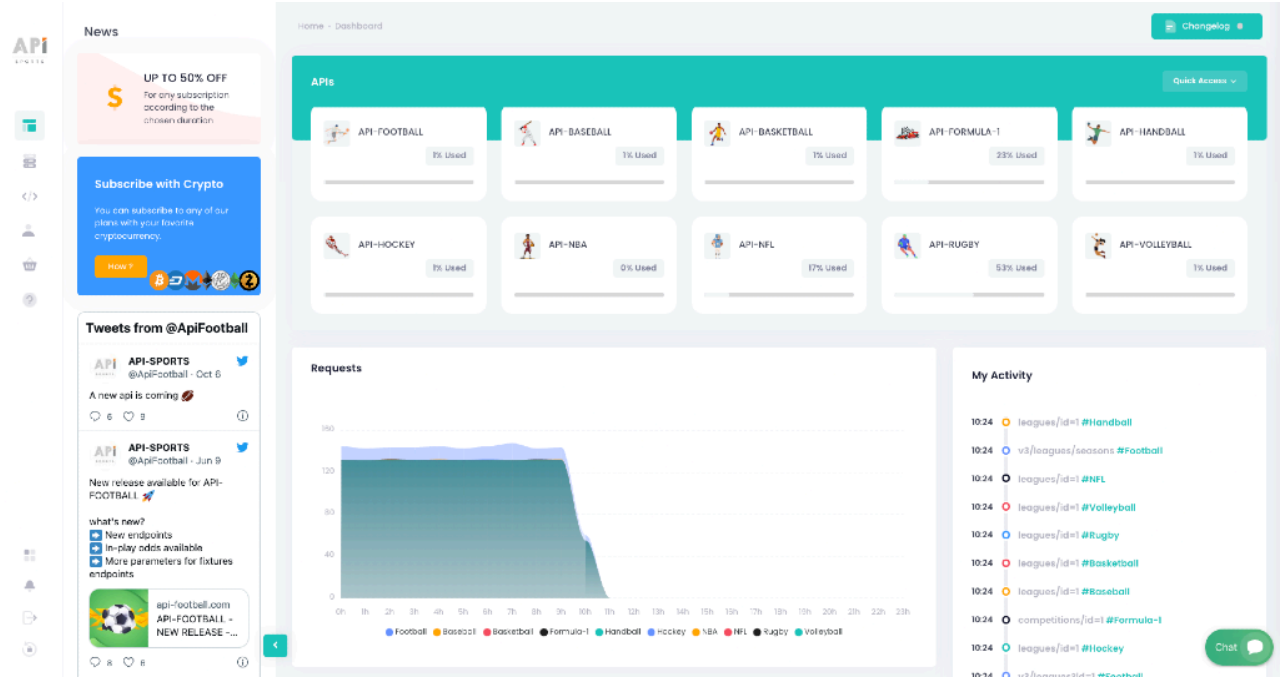
When consuming our API, you will always receive the following headers appended to the response:

- `x-ratelimit-requests-limit` : The number of requests allocated per day according to your subscription.
- `x-ratelimit-requests-remaining` : The number of remaining requests per day according to your subscription.
- `X-RateLimit-Limit` : Maximum number of API calls per minute.
- `X-RateLimit-Remaining` : Number of API calls remaining before reaching the limit per minute.

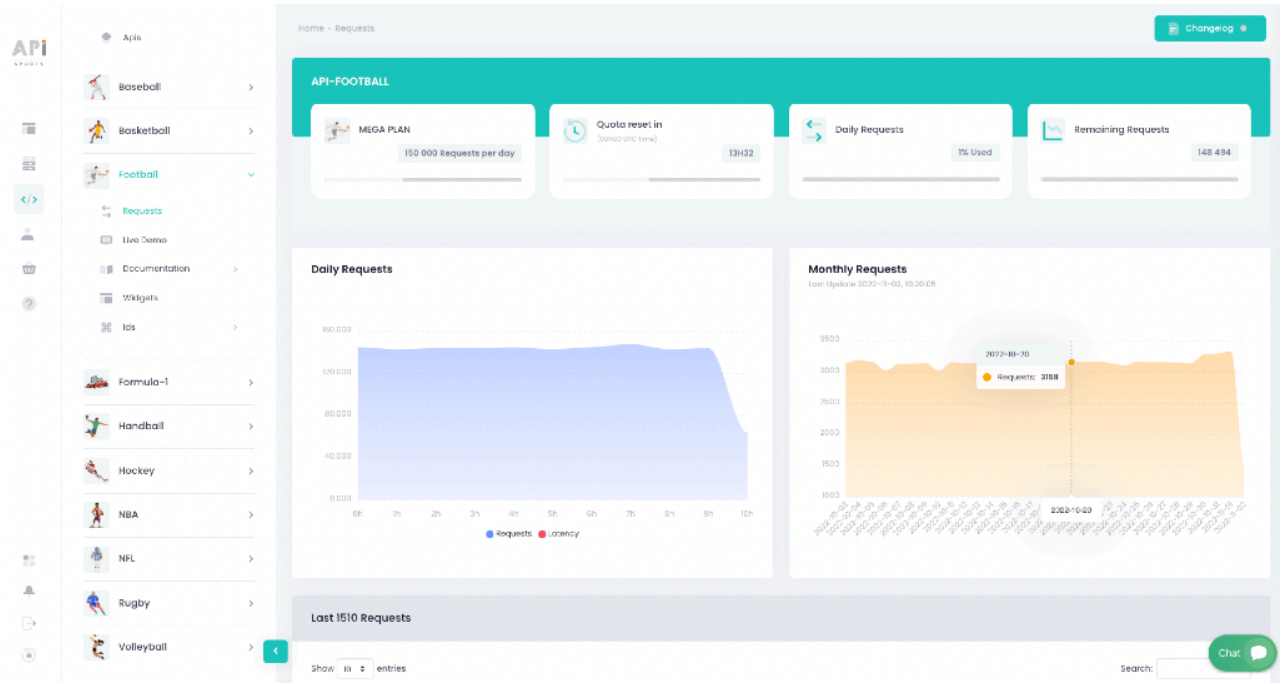
Rate Limiting Policy

If you exceed your allowed request rate per minute, either through continuous excessive usage or by generating abnormal traffic spikes, your access may be temporarily or permanently blocked by our firewall without prior notice. This ensures service stability and fair usage for all customers.

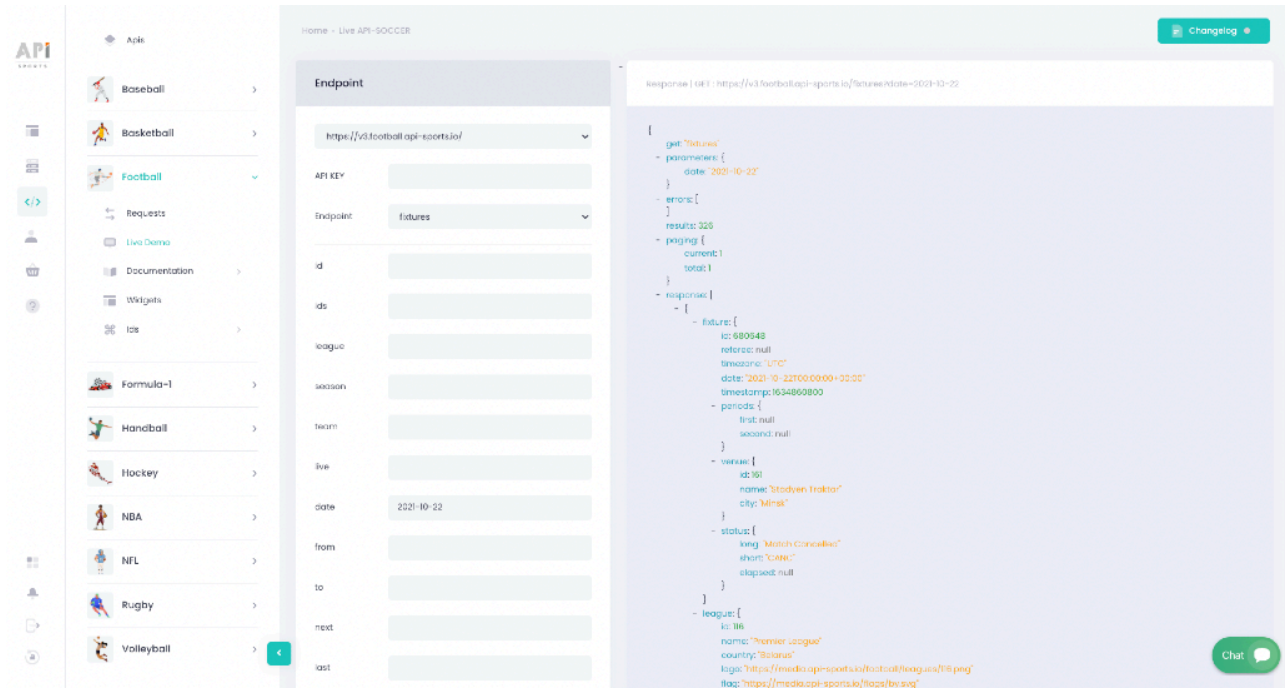
Dashboard



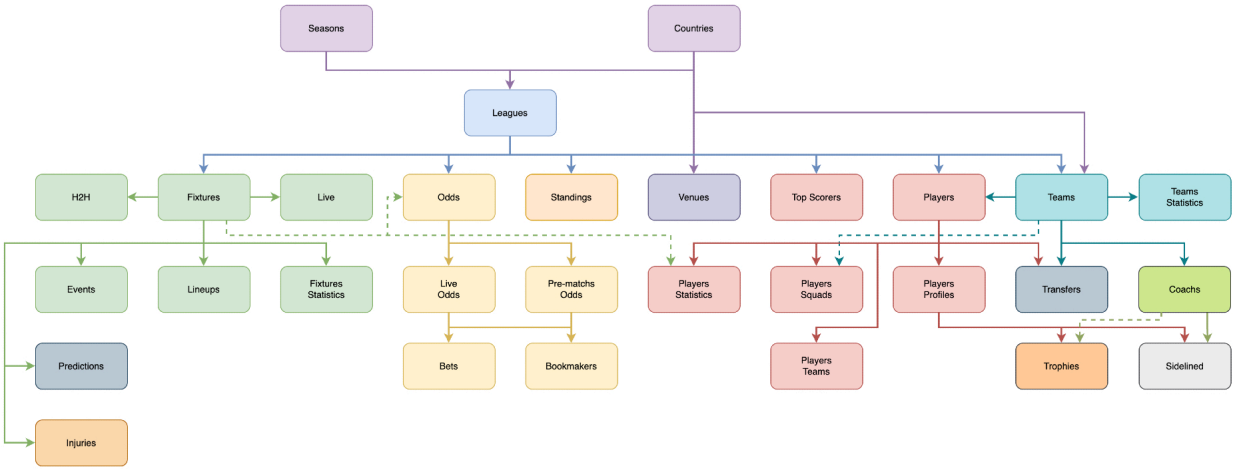
Requests



Live tester



Architecture



Logos / Images

Calls to logos/images do not count towards your daily quota and are provided for free. However these calls are subject to a **rate per second & minute**, it is recommended to save this data on your side in

order not to slow down or impact the user experience of your application or website. For this you can use **CDNs** such as bunny.net.

We have a tutorial available here, which explains how to set up your own media system with **BunnyCDN**.

Logos, images and trademarks delivered through the API are provided solely for identification and descriptive purposes (e.g., identifying leagues, teams, players or venues). We does not own any of these visual assets, and no intellectual property rights are claimed over them. Some images or data may be subject to intellectual property or trademark rights held by third parties (including but not limited to leagues, federations, or clubs). The use of such content in your applications, websites, or products may require additional authorization or licensing from the respective rights holders. You are fully responsible for ensuring that your usage of any logos, images, or branded content complies with applicable laws in your country or the countries where your services are made available. We are not affiliated with, sponsored by, or endorsed by any sports league, federation, or brand featured in the data provided.

Sample Scripts

Here are some examples of how the API is used in the main development languages.

You have to replace `{endpoint}` by the real name of the endpoint you want to call, like `leagues` or `fixtures` for example. In all the sample scripts we will use the `leagues` endpoint as example.

Also you will have to replace `XxXxXxXxXxXxXxXxXxXxXx` with your API-KEY provided in the dashboard.

C

`libcurl`

```
CURL *curl;
CURLcode res;
curl = curl_easy_init();
if(curl) {
  curl_easy_setopt(curl, CURLOPT_CUSTOMREQUEST, "GET");
  curl_easy_setopt(curl, CURLOPT_URL, "https://v3.football.api-sports.io/leagues
  curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1L);
  curl_easy_setopt(curl, CURLOPT_DEFAULT_PROTOCOL, "https");
  struct curl_slist *headers = NULL;
  headers = curl_slist_append(headers, "x-apisports-key: XxXxXxXxXxXxXxXxXxXxXx
```

```
  curl_easy_setopt(curl, CURLOPT_HTTPHEADER, headers);
  res = curl_easy_perform(curl);
}
curl_easy_cleanup(curl);
```

C#

`RestSharp`

```
var client = new RestClient("https://v3.football.api-sports.io/leagues");
client.Timeout = -1;
var request = new RestRequest(Method.GET);
request.AddHeader("x-apisports-key", "XxXxXxXxXxXxXxXxXxXxXxXx");
IRestResponse response = client.Execute(request);
Console.WriteLine(response.Content);
```

cURL

`Curl`

```
curl --request GET \
    --url https://v3.football.api-sports.io/leagues \
    --header 'x-apisports-key: XxXxXxXxXxXxXxXxXxXxXxXx'
```

Dart

`http`

```
var headers = {
  'x-apisports-key': 'XxXxXxXxXxXxXxXxXxXxXxXx',
};
var request = http.Request('GET', Uri.parse('https://v3.football.api-sports.io/l
```

```
request.headers.addAll(headers);

http.StreamedResponse response = await request.send();

if (response.statusCode == 200) {
  print(await response.stream.bytesToString());
}
else {
  print(response.reasonPhrase);
}
```

Go

Native

```
package main

import (
  "fmt"
  "net/http"
  "io/ioutil"
)

func main() {

  url := "https://v3.football.api-sports.io/leagues"
  method := "GET"

  client := &http.Client {
  }
  req, err := http.NewRequest(method, url, nil)

  if err != nil {
    fmt.Println(err)
    return
  }
  req.Header.Add("x-apisports-key", "XxXxXxXxXxXxXxXxXxXxXxXx")

  res, err := client.Do(req)
  if err != nil {
    fmt.Println(err)
    return
```

```
  }
  defer res.Body.Close()

  body, err := ioutil.ReadAll(res.Body)
  if err != nil {
    fmt.Println(err)
    return
  }
  fmt.Println(string(body))
}
```

Java

OkHttp

```
var myHeaders = new Headers();
myHeaders.append("x-apisports-key", "XxXxXxXxXxXxXxXxXxXxXxXx");

var requestOptions = {
  method: 'GET',
  headers: myHeaders,
  redirect: 'follow'
};
```

Unirest

```
Unirest.setTimeouts(0, 0);
HttpResponse<String> response = Unirest.get("https://v3.football.api-sports.io/l
  .header("x-apisports-key", "XxXxXxXxXxXxXxXxXxXxXxXx")
  .asString();
```

Javascript

Fetch

```
var myHeaders = new Headers();
myHeaders.append("x-apisports-key", "XxXxXxXxXxXxXxXxXxXxXxXx");
```

```
var requestOptions = {
  method: 'GET',
  headers: myHeaders,
  redirect: 'follow'
};

fetch("https://v3.football.api-sports.io/leagues", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

jQuery

```
var settings = {
  "url": "https://v3.football.api-sports.io/leagues",
  "method": "GET",
  "timeout": 0,
  "headers": {
    "x-apisports-key": "XxXxXxXxXxXxXxXxXxXxXxXx",
  },
};

$.ajax(settings).done(function (response) {
  console.log(response);
});
```

XHR

```
var xhr = new XMLHttpRequest();
xhr.withCredentials = true;

xhr.addEventListener("readystatechange", function() {
  if(this.readyState === 4) {
    console.log(this.responseText);
  }
});

xhr.open("GET", "https://v3.football.api-sports.io/leagues");
xhr.setRequestHeader("x-apisports-key", "XxXxXxXxXxXxXxXxXxXxXxXx");

xhr.send();
```

NodeJs

Axios

```
var axios = require('axios');

var config = {
  method: 'get',
  url: 'https://v3.football.api-sports.io/leagues',
  headers: {
    'x-apisports-key': 'XxXxXxXxXxXxXxXxXxXxXxXx',
  }
};

axios(config)
.then(function (response) {
  console.log(JSON.stringify(response.data));
})
.catch(function (error) {
  console.log(error);
});
```

Native

```
var https = require('follow-redirects').https;
var fs = require('fs');

var options = {
  'method': 'GET',
  'hostname': 'v3.football.api-sports.io',
  'path': '/leagues',
  'headers': {
    'x-apisports-key': 'XxXxXxXxXxXxXxXxXxXxXxXx',
  },
  'maxRedirects': 20
};

var req = https.request(options, function (res) {
  var chunks = [];

  res.on("data", function (chunk) {
    chunks.push(chunk);
  });
```

```
  res.on("end", function (chunk) {
    var body = Buffer.concat(chunks);
    console.log(body.toString());
  });

  res.on("error", function (error) {
    console.error(error);
  });
});

req.end();
```

Requests

```
var request = require('request');
var options = {
  'method': 'GET',
  'url': 'https://v3.football.api-sports.io/leagues',
  'headers': {
    'x-apisports-key': 'XxXxXxXxXxXxXxXxXxXxXxXx',
  }
};
request(options, function (error, response) {
  if (error) throw new Error(error);
  console.log(response.body);
});
```

Unirest

```
var unirest = require('unirest');
var req = unirest('GET', 'https://v3.football.api-sports.io/leagues')
  .headers({
    'x-apisports-key': 'XxXxXxXxXxXxXxXxXxXxXxXx',
  })
  .end(function (res) {
    if (res.error) throw new Error(res.error);
    console.log(res.raw_body);
  });
```

Objective-c

NSURLSession

```objc
#import <Foundation/Foundation.h>

dispatch_semaphore_t sema = dispatch_semaphore_create(0);

NSMutableURLRequest *request = [NSMutableURLRequest requestWithURL:[NSURL URLWit
  cachePolicy:NSURLRequestUseProtocolCachePolicy
  timeoutInterval:10.0];
NSDictionary *headers = @{
  @"x-apisports-key": @"XxXxXxXxXxXxXxXxXxXxXxXx",
};

[request setAllHTTPHeaderFields:headers];

[request setHTTPMethod:@"GET"];

NSURLSession *session = [NSURLSession sharedSession];
NSURLSessionDataTask *dataTask = [session dataTaskWithRequest:request
completionHandler:^(NSData *data, NSURLResponse *response, NSError *error) {
  if (error) {
    NSLog(@"%@", error);
    dispatch_semaphore_signal(sema);
  } else {
    NSHTTPURLResponse *httpResponse = (NSHTTPURLResponse *) response;
    NSError *parseError = nil;
    NSDictionary *responseDictionary = [NSJSONSerialization JSONObjectWithData:d
    NSLog(@"%@",responseDictionary);
    dispatch_semaphore_signal(sema);
  }
}];
[dataTask resume];
dispatch_semaphore_wait(sema, DISPATCH_TIME_FOREVER);
```

OCaml

`Cohttp`

```ocaml
open Lwt
open Cohttp
open Cohttp_lwt_unix

let reqBody =
  let uri = Uri.of_string "https://v3.football.api-sports.io/leagues" in
  let headers = Header.init ()
```

```
    |> fun h -> Header.add h "x-apisports-key" "XxXxXxXxXxXxXxXxXxXxXxXx"
  in
  Client.call ~headers `GET uri >>= fun (_resp, body) ->
  body |> Cohttp_lwt.Body.to_string >|= fun body -> body

let () =
  let respBody = Lwt_main.run reqBody in
  print_endline (respBody)
```

Php

cURL

```
$curl = curl_init();

curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://v3.football.api-sports.io/leagues',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => '',
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 0,
  CURLOPT_FOLLOWLOCATION => true,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => 'GET',
  CURLOPT_HTTPHEADER => array(
    'x-apisports-key: XxXxXxXxXxXxXxXxXxXxXxXx',
  ),
));

$response = curl_exec($curl);

curl_close($curl);
echo $response;
```

Request2

```
<?php
require_once 'HTTP/Request2.php';
$request = new HTTP_Request2();
$request->setUrl('https://v3.football.api-sports.io/leagues');
$request->setMethod(HTTP_Request2::METHOD_GET);
$request->setConfig(array(
  'follow_redirects' => TRUE
```

```
));
$request->setHeader(array(
  'x-apisports-key' => 'XxXxXxXxXxXxXxXxXxXxXxXx',
));
try {
  $response = $request->send();
  if ($response->getStatus() == 200) {
    echo $response->getBody();
  }
  else {
    echo 'Unexpected HTTP status: ' . $response->getStatus() . ' ' .
    $response->getReasonPhrase();
  }
}
catch(HTTP_Request2_Exception $e) {
  echo 'Error: ' . $e->getMessage();
}
```

Http

```
$client = new http\Client;
$request = new http\Client\Request;
$request->setRequestUrl('https://v3.football.api-sports.io/leagues');
$request->setRequestMethod('GET');
$request->setHeaders(array(
    'x-apisports-key' => 'XxXxXxXxXxXxXxXxXxXxXxXx'
));
$client->enqueue($request)->send();
$response = $client->getResponse();
echo $response->getBody();
```

PowerShell

RestMethod

```
$headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
$headers.Add("x-apisports-key", "XxXxXxXxXxXxXxXxXxXxXxXx")

$response = Invoke-RestMethod 'https://v3.football.api-sports.io/leagues' -Metho
$response | ConvertTo-Json
```

Python

`http.client`

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/leagues", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`Requests`

```
url = "https://v3.football.api-sports.io/leagues"

payload={}
headers = {
  'x-apisports-key': 'XxXxXxXxXxXxXxXxXxXxXxXx',
}

response = requests.request("GET", url, headers=headers, data=payload)

print(response.text)
```

Ruby

`Net::HTTP`

```ruby
require 'uri'
require 'net/http'
require 'openssl'

url = URI("https://v3.football.api-sports.io/leagues")

http = Net::HTTP.new(url.host, url.port)
http.use_ssl = true
http.verify_mode = OpenSSL::SSL::VERIFY_NONE

request = Net::HTTP::Get.new(url)
request["x-apisports-key"] = 'XxXxXxXxXxXxXxXxXxXxXxXx'

response = http.request(request)
puts response.read_body
```

Shell

Httpie

```shell
http --follow --timeout 3600 GET 'https://v3.football.api-sports.io/leagues' \
 x-apisports-key:'XxXxXxXxXxXxXxXxXxXxXxXx' \
```

wget

```shell
wget --no-check-certificate --quiet \
  --method GET \
  --timeout=0 \
  --header 'x-apisports-key: XxXxXxXxXxXxXxXxXxXxXxXx' \
   'https://v3.football.api-sports.io/leagues'
```

Swift

URLSession

```
import Foundation
#if canImport(FoundationNetworking)
import FoundationNetworking
#endif

var semaphore = DispatchSemaphore (value: 0)

var request = URLRequest(url: URL(string: "https://v3.football.api-sports.io/lea
request.addValue("XxXxXxXxXxXxXxXxXxXxXxXx", forHTTPHeaderField: "x-apisports-ke

request.httpMethod = "GET"

let task = URLSession.shared.dataTask(with: request) { data, response, error in
  guard let data = data else {
    print(String(describing: error))
    semaphore.signal()
    return
  }
  print(String(data: data, encoding: .utf8)!)
  semaphore.signal()
}

task.resume()
semaphore.wait()
```

Changelog

3.9.3

- Add endpoint `players/profiles` that returns the list of all available players
- Add endpoint `players/teams` that returns the list of teams and seasons in which the player played during his career
- Endpoint `fixtures`
  - Add field `extra` that returns the additional time played in a half
  - Add field `standings` indicating whether the fixture's competition covers standings (True | False)
- Endpoint `fixtures/rounds`
  - Add the `dates` parameter that allows to retrieve the dates of each round in the response

- Endpoint `fixtures/statistics`
  - Add the `half` parameter that allows to retrieve the halftime statistics in the response
- Endpoint `injuries`
  - Add the `ids` parameter that allows to retrieve data from several fixtures in one call
- Endpoint `teams/statistics`, more statistics added
  - Goals Over
  - Goals Under
- Endpoint `sidelined`
  - Add the `players` and `coachs` parameters that allows to retrieve data from several players/coachs in one call
- Endpoint `trophies`
  - Add the `players` and `coachs` parameters that allows to retrieve data from several players/coachs in one call

3.9.2

- Endpoint `odds`
  - Add endpoint `odds/live`
  - Add endpoint `odds/live/bets`
- Endpoint `teams`
  - Add parameter `code`
  - Add parameter `venue`
  - Add endpoint `teams/countries`
- Endpoint `fixtures`
  - Add the `ids` parameter that allows to retrieve data from several fixtures including events, lineups, statistics and players in one Api call
  - Add the Possibility to add several status for the `status` parameter
  - Add parameter `venue`
- Endpoint `fixtures/headtohead`
  - Add the Possibility to add several status for the `status` parameter
  - Add parameter `venue`

3.8.1

- Add endpoint `injuries`
- Add endpoint `players/squads`
- Add endpoint `players/topassists`
- Add endpoint `players/topyellowcards`

- Add endpoint `players/topredcards`
- Endpoint `fixtures/lineups`
  - Add players positions on the grid
  - Add players' jerseys colors
- Endpoint `fixtures/events`
  - add VAR events
- Endpoint `teams`
  - Add tri-code
- Endpoint `teams/statistics`, more statistics added
  - Scoring minute
  - Cards per minute
  - Most played formation
  - Penalty statistics
- Add Coaches Photos

CDN

Optimizing Sports Websites with BunnyCDN

BunnyCDN is a Content Delivery Network *(CDN)* that delivers a global content distribution experience. With strategically positioned servers, BunnyCDN ensures swift and reliable delivery of static content, optimizing website performance with features like intelligent image optimization, sophisticated caching, and advanced security measures.

Unlocking Media Delivery Excellence with BunnyCDN:

- **Quick Configuration:** Set up your media CDN in just 5 minutes. Define cache times, customize your domain – it's that simple.
- **Global Accessibility:** Leverage BunnyCDN's expansive server network for swift and dependable content delivery worldwide.
- **Customized Configuration:** Tailor caching, define cache times, and implement CORS headers to create an efficient and seamless user experience.
- **Own Your Domain:** Personalize your media delivery with your domain, enhancing your brand's online presence.
- **Robust Security:** BunnyCDN integrates advanced security features, guaranteeing a secure environment for delivering your content.
- **Responsive Performance:** Experience responsive performance without the need for prior media downloads. Discover the capabilities of BunnyCDN for optimized media delivery.

A tutorial is available here on our blog to help you configure it.

Databases Solutions

Enhance Your Data Management with Aiven

Integrating databases into your application can greatly enhance data management and storage. If you're looking for high-performing, flexible, and secure database solutions, we recommend checking out Aiven.

Aiven is a cloud platform that offers a range of managed database services, including relational databases, NoSQL databases, streaming data processing systems, and much more. Their offerings include `PostgreSQL`, `MySQL`, `Cassandra`, `Redis`, `Kafka`, and many other databases, all with simplified management, high availability, and advanced security.

Moreover, **Aiven** provides a free tier to get started, along with testing credits to explore their offerings. This opportunity allows you to evaluate their platform and determine if it meets your needs.

One particularly attractive feature of **Aiven** is that they work with multiple cloud providers, including `Google Cloud`, `Amazon Web Services (AWS)`, `Microsoft Azure`, `DigitalOcean`, and more. This means you have the flexibility to choose the best cloud infrastructure for your project.

In terms of reliability, **Aiven** is committed to providing a **99.99%** Service Level Agreement (SLA), ensuring continuous and highly available service.

- To test their services, visit this page.
- If you're a developer, explore their DEV center for technical information.
- Check out Aiven's documentation for detailed information on their services and features.

By integrating **Aiven** with our API, you can efficiently store, manage, and analyze your data while taking advantage of their cloud database solutions' flexibility and scalability.

Real-Time Data Management with Firebase

When you're looking for a real-time data management solution for your application, Firebase's Realtime Database is a powerful choice. Explore how Firebase can enhance real-time data management for your application.

Firebase's Realtime Database offers a cloud-based real-time database that synchronizes data in real-time across users and devices. This makes it an ideal choice for applications that require instant data updates.

Why Choose Firebase's Realtime Database?

- **Real-Time Data:** Firebase allows you to store real-time data, meaning that updates are instantly propagated to all connected users.
- **Easy Synchronization:** Data is automatically synchronized across all devices, providing a consistent and real-time user experience.
- **Built-In Security:** Firebase offers flexible security rules to control data access and ensure privacy.
- **Simplified Integration:** Firebase's Realtime Database easily integrates with other Firebase services, simplifying backend management.

Helpful Links:

- Explore Firebase's Realtime Database: Discover the features and advantages of Firebase's Realtime Database for efficient real-time data management.
- Firebase's Realtime Database Documentation: Refer to the comprehensive documentation for Firebase's Realtime Database for a smooth integration.

A tutorial describing each step is available on our blog here.

Widgets

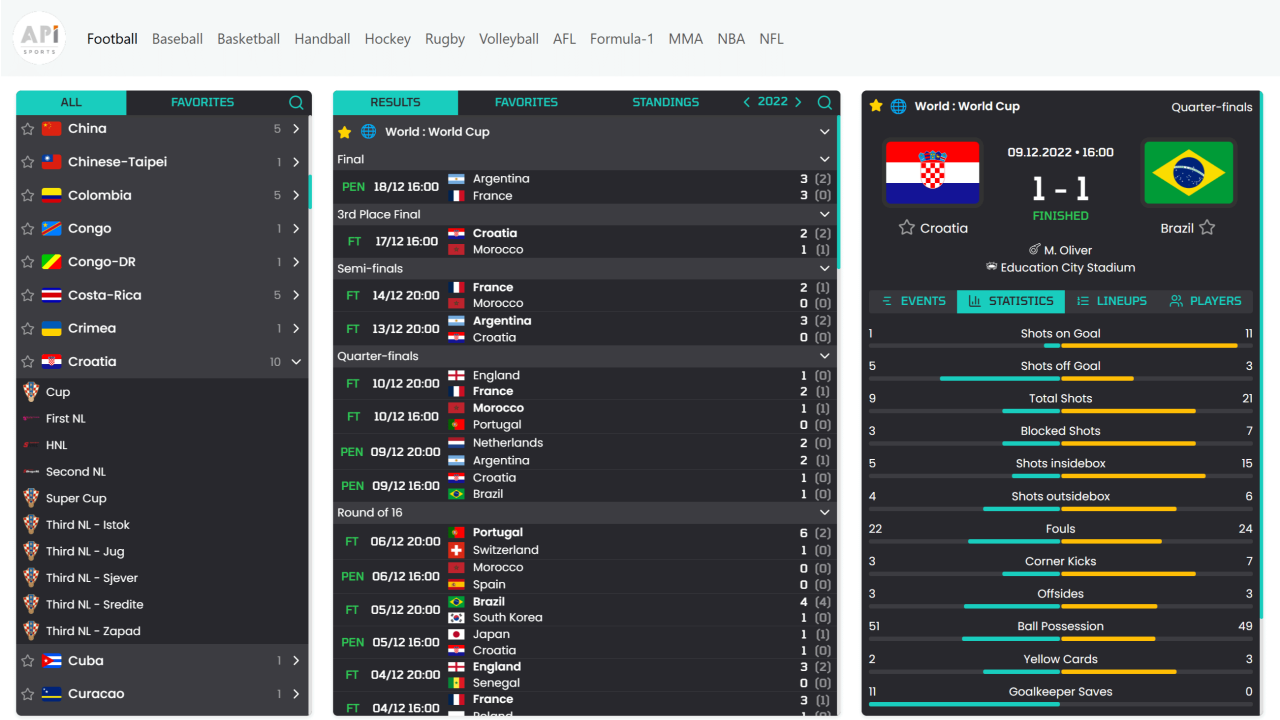
API-SPORTS widgets allow you to easily display **dynamic sports data** on your website.

They are designed to be:

- **Ultra-modular**: each component is autonomous
- **Customisable**: language, theme, content, behaviour
- **Easy to integrate**: no framework required, a simple HTML tag is all you need

They use request from your API-SPORTS account and work with **all plans**, including the free plan.

Find all the documentation on widgets here



Security

Our widgets use your account's API-KEY, which must be specified in the `data-key` attribute of your widget configuration.

When using these widgets it is important to be aware that your API-KEY will be visible to the users of your site, it is possible to protect yourself from this by allowing only the desired domains or IP in our dashboard. This way no one else can use your API-KEY for you. If you have already set up your widget and have not activated this option, you can reset your API-KEY and activate this option after.

You can further enhance security by completely hiding your API-KEY from the source code by following this tutorial.

Caching Data

By using Widgets, each visit to a page on your website triggers one or more API requests to retrieve data. Without a caching system, your daily quota can be reached very quickly.

Example: If a page triggers a single API request per visitor and you receive 80 visits to that page in one minute, this results in **80 API requests**. Over a full day, that can add up to **115 200 requests**.

By implementing a caching system, even with a very short duration, such as **60 seconds**, you can drastically reduce the number of requests. The first visit will trigger an API request, but the response will then be cached for the next 60 seconds. This means that if 80 visitors access the same page

within that time frame, **only the first request** will reach the API, while the next 79 will be served directly from the cache.

With this system in place, you reduce usage from **115 200** requests per day to just **1 440**.

A full tutorial is available here, explaining step by step how to set up an effective caching system.

Debugging

If the widget does not display the requested information, it is possible to set the `data-show-errors` tag to **true** to display error messages directly in the widget and in the console. This can be due to several things like : *(Non-exhaustive list)*

- You have reached your daily number of requests
- Tags are incorrectly filled in
- Your API-KEY is incorrect

All available widgets

Below is a list of all available widgets:

- `games` → list of matches
- `game` → details of a match
- `team` → team profile
- `player` → player profile
- `standings` → league table
- `league` → schedule
- `leagues` → list of all leagues
- `h2h` → historical head-to-head
- `races`, `race`, `driver` → Formula 1
- `fights`, `fight`, `fighter` → MMA

> Each widget adapts automatically based on the selected sport.

Before You Begin

Dynamic targeting

Some widgets, such as `games`, can dynamically open other widgets like `game`, `standings`, `player`, and more.
This interaction is enabled using the `data-target-*` attributes.

These attributes allow you to define **where** the opened widget should be rendered:

- `modal` → renders the widget inside a modal.
- CSS selector (`#id` or `.class`) → injects the widget into a specific HTML element on the page.

These targeting options are available for:

General sports widgets (Football, Basketball, etc.):

- `data-target-game`
- `data-target-standings`
- `data-target-team`
- `data-target-player`
- `data-target-league`

Formula 1 specific:

- `data-target-race`
- `data-target-ranking`
- `data-target-driver`

MMA specific:

- `data-target-fight`
- `data-target-fighter`

Target a container by ID

```
<api-sports-widget data-type="games"></api-sports-widget>

<div id="details"></div>

<api-sports-widget
  data-type="config"
  data-key="Your-Api-Key-Here"
  data-sport="football"
  data-target-game="#details"
></api-sports-widget>
```

Target using modal

```
  <api-sports-widget data-type="games"></api-sports-widget>

  <api-sports-widget
    data-type="config"
    data-key="Your-Api-Key-Here"
    data-sport="football"
    data-target-game="modal"
  ></api-sports-widget>
```

Language

The `data-lang` attribute allows you to easily switch the interface language of all widgets.

Available languages:

- `en` (English)
- `fr` (French)
- `es` (Spanish)
- `it` (Italian)

Example usage

```
<api-sports-widget
  data-type="config"
  data-key="Your-Api-Key"
  data-sport="football"
  data-lang="en"
  data-custom-lang="https://yourdomain.com/lang/en.json"
></api-sports-widget>
```

> Example
>
> 
>

Custom translations:

For complete control over wording, you can load your own translation file using `data-custom-lang`. This file must be a valid JSON object following the internal key structure.

You can download the translation file here.

It allows you to:

- Override specific labels
- Translate missing terms
- Adapt terminology to your audience

Example JSON format:

```
{
  "all": "All",
  "live": "Now Live",
  "finished": "Completed",
  "scheduled": "Coming Up",
  "favorites": "Favorites",
}
```

> You can use `data-lang` and `data-custom-lang` **together**.
> If a key is defined in both, the **custom file will take priority**.

Exemple for custom translation

```
  <api-sports-widget data-type="games" data-target-game="modal"></api-sports-wid

  <api-sports-widget
```

```
    data-type="config"
    data-key="Your-Api-Key-Here"
    data-sport="football"
    data-lang="custom"
    data-custom-lang="https://yourdomain.com/lang/custom.json"
  ></api-sports-widget>
```

You have a tutorial available here

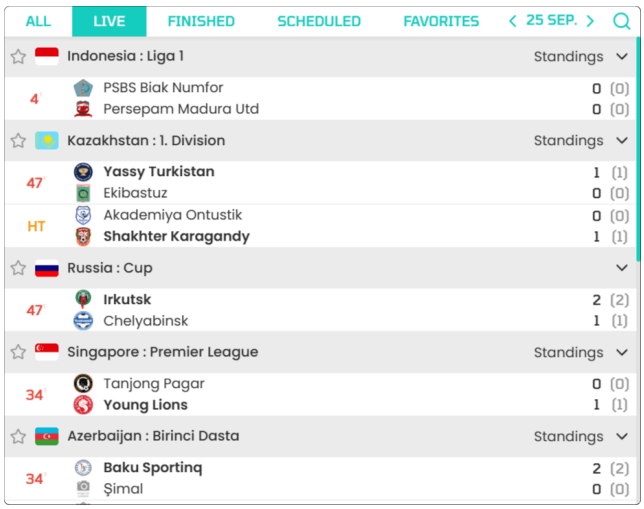
Predefined themes

Four built-in themes are available by default. You can set them using the `data-theme` attribute on any widget.

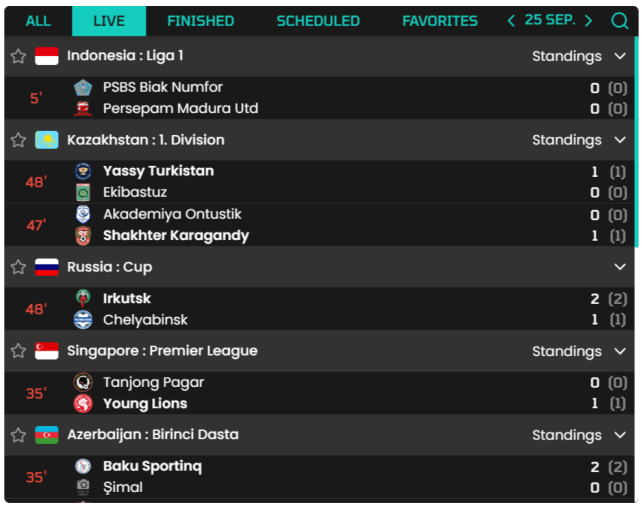
- `white` (default)
- `grey`
- `dark`
- `blue`

Each theme adjusts background colors, text colors, button styles, borders, and more.

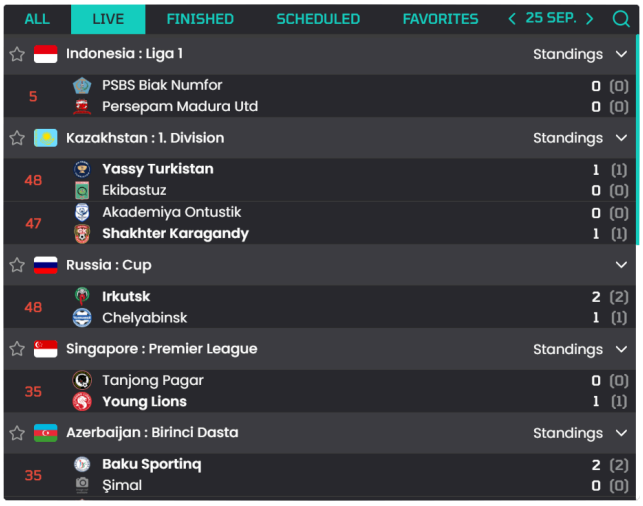
White



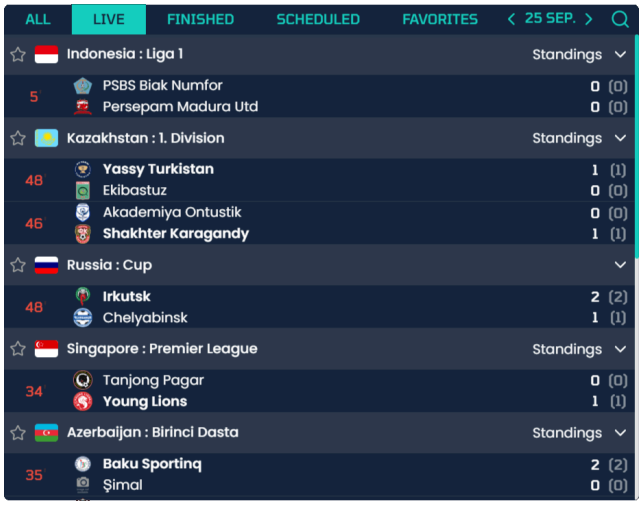
Dark



Grey



Blue



Custom theme

You can override the default styles by creating your own CSS theme using the `data-theme` attribute and custom variable declarations.

Example:

```
api-sports-widget[data-theme="MyTheme"] {
  --primary-color: #18cfc0;
  --success-color: #2ecc58;
  --warning-color: #f39c12;
  --danger-color: #e74c3c;
  --light-color: #898989;

  --home-color: var(--primary-color);
  --away-color: #ffc107;

  --text-color: #333;
  --text-color-info: #333;

  --background-color: #fff;

  --primary-font-size: 0.72rem;
  --secondary-font-size: 0.75rem;
  --button-font-size: 0.8rem;
  --title-font-size: 0.9rem;

  --header-text-transform: uppercase;
  --button-text-transform: uppercase;
  --title-text-transform: uppercase;
```

```
  --border: 1px solid #95959530;
  --game-height: 2.3rem;
  --league-height: 2.35rem;

  --score-size: 2.25rem;
  --flag-size: 22px;
  --teams-logo-size: 18px;
  --teams-logo-size-xl: 5rem;
  --hover: rgba(200, 200, 200, 0.15);
}
```

```
    <api-sports-widget data-type="games"></api-sports-widget>

    <div id="game-container"></div>

    <api-sports-widget
      data-type="config"
      data-key="Your-Api-Key-Here"
      data-sport="football"
      data-theme="MyTheme"
    ></api-sports-widget>
```

Find all the documentation on widgets here

Timezone

Timezone

Get the list of available timezone to be used in the fixtures endpoint.

> This endpoint does not require any parameters.

Update Frequency : This endpoint contains all the existing timezone, it is not updated.

Recommended Calls : 1 call when you need.

HEADER PARAMETERS

x-apisports-key required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /timezone

Request samples

Php | Python | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/timezone", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all

```
{
    "get": "timezone",
    "parameters": [ ],
    "errors": [ ],
    "results": 425,
  - "paging": {
        "current": 1,
        "total": 1
    },
  - "response": [
        "Africa/Abidjan",
        "Africa/Accra",
        "Africa/Addis_Ababa",
        "Africa/Algiers",
        "Africa/Asmara"
    ]
}
```


Countries

Countries

Get the list of available countries for the `leagues` endpoint.

The `name` and `code` fields can be used in other endpoints as filters.

To get the flag of a country you have to call the following url: `https://media.api-sports.io/flags/{country_code}.svg`

> Examples available in Request samples "Use Cases".

All the parameters of this endpoint can be used together.

Update Frequency : This endpoint is updated each time a new league from a country not covered by the API is added.

Recommended Calls : 1 call per day.

QUERY PARAMETERS

name	string The name of the country
code	string [2 .. 6] characters FR, GB-ENG, IT... The Alpha code of the country
search	string = 3 characters The name of the country

HEADER PARAMETERS

x-apisports-key required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /countries

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/countries", headers=headers)
```

```
res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "countries",
  - "parameters": {
        "name": "england"
    },
    "errors": [ ],
    "results": 1,
  - "paging": {
        "current": 1,
        "total": 1
    },
  - "response": [
      + { … }
    ]
}
```

Leagues

Leagues

Get the list of available leagues and cups.

The league `id` are **unique** in the API and leagues keep it across all `seasons`

To get the logo of a competition you have to call the following url: `https://media.api-sports.io/football/leagues/{league_id}.png`

This endpoint also returns the `coverage` of each competition, which makes it possible to know what is available for that league or cup.

The values returned by the coverage indicate the **data available at the moment** you call the API, so for a competition that has not yet started, it is normal to have all the features set to `False`. This will be updated once the competition has started.

> You can find all the leagues ids on our Dashboard.

Example :


```
"coverage": {
  "fixtures": {
      "events": true,
      "lineups": true,
      "statistics_fixtures": false,
      "statistics_players": false
  },
  "standings": true,
  "players": true,
  "top_scorers": true,
  "top_assists": true,
  "top_cards": true,
  "injuries": true,
  "predictions": true,
  "odds": false
}
```


In this example we can deduce that the competition does not have the following features: `statistics_fixtures`, `statistics_players`, `odds` because it is set to `False`.

The coverage of a competition can vary from season to season and values set to `True` do not guarantee 100% data availability.

Some competitions, such as the `friendlies`, are exceptions to the coverage indicated in the `leagues` endpoint, and the data available may differ depending on the match, including livescore, events, lineups, statistics and players.

Competitions are automatically renewed by the API when a new season is available. There may be a delay between the announcement of the official calendar and the availability of data in the API.

For `Cup` competitions, fixtures are automatically added when the two participating teams are known. For example if the current phase is the 8th final, the quarter final will be added once the teams playing this phase are known.

> Examples available in Request samples "Use Cases".

> Most of the parameters of this endpoint can be used together.

Update Frequency : This endpoint is updated several times a day.

Recommended Calls : 1 call per hour.

QUERY PARAMETERS

id	integer The id of the league
name	string The name of the league
country	string The country name of the league
code	string [2 .. 6] characters FR, GB-ENG, IT... The Alpha code of the country
season	integer = 4 characters YYYY The season of the league
team	integer The id of the team
type	string Enum: "league" "cup" The type of the league
current	string Return the list of active seasons or the las... Show pattern Enum: "true" "false" The state of the league
search	string >= 3 characters The name or the country of the league
last	integer <= 2 characters The X last leagues/cups added in the API

HEADER PARAMETERS

x-apisports-key required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

```
GET  /leagues
```

Request samples

Use Cases | Php | **Python** | Node | JavaScript | Curl | Ruby

Copy

```python
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/leagues", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | **204** | **499** | **500**

Content type
application/json

Copy Expand all Collapse all


```
{
   "get": "leagues",
 - "parameters": {
       "id": "39"
   },
   "errors": [ ],
   "results": 1,
```

```
- "paging": {
      "current": 1,
      "total": 1
  },
- "response": [
    + { ... }
  ]
}
```


Seasons

Get the list of available seasons.

All seasons are only **4-digit keys**, so for a league whose season is `2018-2019` like the English Premier League (EPL), the `2018-2019` season in the API will be `2018`.

All `seasons` can be used in other endpoints as filters.

> This endpoint does not require any parameters.

Update Frequency : This endpoint is updated each time a new league is added.

Recommended Calls : 1 call per day.

HEADER PARAMETERS

`x-apisports-key` required	string Your Api-Key

Responses

- **200** OK
- **204** No Content
- **499** Time Out
- **500** Internal Server Error

GET /leagues/seasons

Request samples

Php | Python | Node | JavaScript | Curl | Ruby

Copy

```python
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/leagues/seasons", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "leagues/seasons",
    "parameters": [ ],
    "errors": [ ],
    "results": 12,
  - "paging": {
        "current": 1,
        "total": 1
    },
```

```
  - "response": [
        2008,
        2010,
        2011,
        2012,
        2013,
        2014,
        2015,
        2016,
        2017,
        2018,
        2019,
        2020
    ]
}
```

Teams

Teams information

Get the list of available teams.

The team `id` are **unique** in the API and teams keep it among all the leagues/cups in which they participate.

To get the logo of a team you have to call the following url: `https://media.api-sports.io/football/teams/{team_id}.png`

> You can find all the teams ids on our Dashboard.

> Examples available in Request samples "Use Cases".

> All the parameters of this endpoint can be used together.

This endpoint requires at least one parameter.

Update Frequency : This endpoint is updated several times a week.

Recommended Calls : 1 call per day.

Tutorials :

- HOW TO GET ALL TEAMS AND PLAYERS FROM A LEAGUE ID

QUERY PARAMETERS

id	integer The id of the team
name	string The name of the team
league	integer The id of the league
season	integer = 4 characters YYYY The season of the league
country	string The country name of the team
code	string = 3 characters The code of the team
venue	integer The id of the venue
search	string >= 3 characters The name or the country name of the team

HEADER PARAMETERS

x-apisports-key required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /teams

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```python
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/teams?id=33", headers=headers)

res = conn.getresponse()
data = res.read()
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
   "get": "teams",
 - "parameters": {
       "id": "33"
   },
   "errors": [ ],
   "results": 1,
 - "paging": {
       "current": 1,
       "total": 1
   },
 - "response": [
     + { ... }
   ]
}
```

Teams statistics

Returns the statistics of a team in relation to a given competition and season.

It is possible to add the `date` parameter to calculate statistics from the beginning of the season to the given date. By default the API returns the statistics of all games played by the team for the competition and the season.

Update Frequency : This endpoint is updated twice a day.

Recommended Calls : 1 call per day for the teams who have at least one fixture during the day otherwise 1 call per week.

> Here is an example of what can be achieved

	LIVERPOOL			MANCHESTER UNITED		
	HOME	AWAY	ALL	HOME	AWAY	ALL
Games played	11	10	21	11	11	22
Wins	11	9	20	6	3	9
Draws	0	1	1	4	3	7
Loss	0	0	0	1	5	6
GOALS						
Goals For	29	21	50	24	12	36
Goals Against	9	5	14	10	15	25
GOALS AVERAGE						
Goals For	2.6	2.1	2.4	2.2	1.1	1.6
Goals Against	0.8	0.5	0.7	0.9	1.4	1.1

QUERY PARAMETERS

`league` `required`
integer
The id of the league

`season` `required`
integer = 4 characters YYYY
The season of the league

`team` `required`
integer
The id of the team

`date`
stringYYYY-MM-DD
The limit date

HEADER PARAMETERS

x-apisports-key required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /teams/statistics

Request samples

Use Cases | Php | **Python** | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/teams/statistics?season=2019&team=33&league=39", header

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | **204** | **499** | **500**

Content type
application/json

Copy Expand all Collapse all


```
{
   "get": "teams/statistics",
 - "parameters": {
       "league": "39",
       "season": "2019",
       "team": "33"
   },
   "errors": [ ],
   "results": 11,
 - "paging": {
       "current": 1,
       "total": 1
   },
 - "response": {
     + "league": { ··· },
     + "team": { ··· },
       "form": "WDLDWLDLDWLWDDWWDLWWLWLLDWWDWDWWWWDWDW",
     + "fixtures": { ··· },
     + "goals": { ··· },
     + "biggest": { ··· },
     + "clean_sheet": { ··· },
     + "failed_to_score": { ··· },
     + "penalty": { ··· },
     + "lineups": [ ··· ],
     + "cards": { ··· }
   }
}
```

Teams seasons

Get the list of seasons available for a team.

> Examples available in Request samples "Use Cases".

This endpoint requires at least one parameter.

Update Frequency : This endpoint is updated several times a week.

Recommended Calls : 1 call per day.

QUERY PARAMETERS

team required	integer The id of the team

HEADER PARAMETERS

x-apisports-key required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /teams/seasons

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/teams/seasons?team=33", headers=headers)

res = conn.getresponse()
data = res.read()
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
  "get": "teams/seasons",
- "parameters": {
    "team": "33"
  },
  "errors": [ ],
  "results": 1,
- "paging": {
    "current": 1,
    "total": 1
  },
- "response": [
    2010,
    2011,
    2012,
    2013,
    2014,
    2015,
    2016,
    2017,
    2018,
    2019,
    2020,
    2021
  ]
}
```

Teams countries

Get the list of countries available for the `teams` endpoint.

Update Frequency : This endpoint is updated several times a week.

Recommended Calls : 1 call per day.

HEADER PARAMETERS

x-apisports-key required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /teams/countries

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/teams/countries", headers=headers)

res = conn.getresponse()
data = res.read()
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
   "get": "teams/countries",
   "parameters": [ ],
   "errors": [ ],
   "results": 258,
 - "paging": {
       "current": 1,
       "total": 1
   },
 - "response": [
     + { ... }
   ]
}
```


Venues

Venues

Get the list of available venues.

The venue `id` are **unique** in the API.

To get the image of a venue you have to call the following url: `https://media.api-sports.io/football/venues/{venue_id}.png`

> Examples available in Request samples "Use Cases".

> All the parameters of this endpoint can be used together.

This endpoint requires at least one parameter.

Update Frequency : This endpoint is updated several times a week.

Recommended Calls : 1 call per day.

QUERY PARAMETERS

id	integer The id of the venue
name	string The name of the venue
city	string The city of the venue
country	string The country name of the venue
search	string >= 3 characters The name, city or the country of the venue

HEADER PARAMETERS

x-apisports-key required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /venues

Request samples

Use Cases | Php | **Python** | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/venues?id=556", headers=headers)

res = conn.getresponse()
data = res.read()
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
   "get": "venues",
 - "parameters": {
       "id": "556"
   },
   "errors": [ ],
   "results": 1,
 - "paging": {
       "current": 1,
       "total": 1
   },
 - "response": [
     + { … }
   ]
}
```

Standings

Standings

Get the standings for a league or a team.

Return a table of one or more rankings according to the league / cup.

Some competitions have several rankings in a year, group phase, opening ranking, closing ranking etc...

> Examples available in Request samples "Use Cases".

> Most of the parameters of this endpoint can be used together.

Update Frequency : This endpoint is updated every hour.

Recommended Calls : 1 call per hour for the leagues or teams who have at least one fixture in progress otherwise 1 call per day.

Tutorials :

- HOW TO GET STANDINGS FOR ALL CURRENT SEASONS

QUERY PARAMETERS

`league`	integer The id of the league
`season` required	integer = 4 characters YYYY The season of the league
`team`	integer The id of the team

HEADER PARAMETERS

`x-apisports-key` required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

`GET /standings`

Request samples

Use Cases | Php | **Python** | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/standings?league=39&season=2019", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | **204** | **499** | **500**

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "standings",
  - "parameters": {
        "league": "39",
        "season": "2019"
    },
    "errors": [ ],
    "results": 1,
```

```
- "paging": {
    "current": 1,
    "total": 1
  },
- "response": [
   + { ... }
  ]
}
```


Fixtures

Rounds

Get the rounds for a league or a cup.

The `round` can be used in endpoint fixtures as filters

> Examples available in Request samples "Use Cases".

Update Frequency : This endpoint is updated every day.

Recommended Calls : 1 call per day.

QUERY PARAMETERS

`league` required	integer The id of the league
`season` required	integer `= 4 characters` YYYY The season of the league
`current`	boolean Enum: `"true"` `"false"` The current round only
`dates`	boolean Default: `false` Enum: `"true"` `"false"` Add the dates of each round in the response

`timezone`	string A valid timezone from the endpoint `Timezone`

HEADER PARAMETERS

`x-apisports-key` required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

`GET` `/fixtures/rounds`

Request samples

Use Cases | Php | **Python** | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/fixtures/rounds?season=2019&league=61", headers=headers

res = conn.getresponse()
data = res.read()
```

```
print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Example
Default

Copy Expand all Collapse all


```
{
    "get": "fixtures/rounds",
  - "parameters": {
        "league": "39",
        "season": "2019"
    },
    "errors": [ ],
    "results": 38,
  - "paging": {
        "current": 1,
        "total": 1
    },
```

```
- "response": [
        "Regular Season - 1",
        "Regular Season - 2",
        "Regular Season - 3",
        "Regular Season - 4",
        "Regular Season - 5",
        "Regular Season - 6",
        "Regular Season - 7",
        "Regular Season - 8",
        "Regular Season - 9",
        "Regular Season - 10",
        "Regular Season - 11",
        "Regular Season - 12",
        "Regular Season - 13",
        "Regular Season - 14",
        "Regular Season - 15",
        "Regular Season - 16",
        "Regular Season - 17",
        "Regular Season - 18",
        "Regular Season - 18",
        "Regular Season - 19",
        "Regular Season - 20",
        "Regular Season - 21",
        "Regular Season - 22",
        "Regular Season - 23",
        "Regular Season - 24",
        "Regular Season - 25",
        "Regular Season - 26",
        "Regular Season - 27",
        "Regular Season - 28",
        "Regular Season - 29",
        "Regular Season - 30",
        "Regular Season - 31",
        "Regular Season - 32",
        "Regular Season - 33",
        "Regular Season - 34",
        "Regular Season - 35",
        "Regular Season - 36",
        "Regular Season - 37",
        "Regular Season - 38"
    ]
}
```

Fixtures

For all requests to fixtures you can add the query parameter `timezone` to your request in order to retrieve the list of matches in the time zone of your choice like *"Europe/London"*

To know the list of available time zones you have to use the endpoint timezone.

Available fixtures status

SHORT	LONG	TYPE	DESCRIPTION
TBD	Time To Be Defined	Scheduled	Scheduled but date and time are not known
NS	Not Started	Scheduled	
1H	First Half, Kick Off	In Play	First half in play
HT	Halftime	In Play	Finished in the regular time
2H	Second Half, 2nd Half Started	In Play	Second half in play
ET	Extra Time	In Play	Extra time in play
BT	Break Time	In Play	Break during extra time
P	Penalty In Progress	In Play	Penaly played after extra time
SUSP	Match Suspended	In Play	Suspended by referee's decision, may be rescheduled another day
INT	Match Interrupted	In Play	Interrupted by referee's decision, should resume in a few minutes
FT	Match Finished	Finished	Finished in the regular time
AET	Match Finished	Finished	Finished after extra time without going to the penalty shootout
PEN	Match Finished	Finished	Finished after the penalty shootout
PST	Match Postponed	Postponed	Postponed to another day, once the new date and time is known the status will change to Not Started
CANC	Match Cancelled	Cancelled	Cancelled, match will not be played

SHORT	LONG	TYPE	DESCRIPTION
ABD	Match Abandoned	Abandoned	Abandoned for various reasons (Bad Weather, Safety, Floodlights, Playing Staff Or Referees), Can be rescheduled or not, it depends on the competition
AWD	Technical Loss	Not Played	
WO	WalkOver	Not Played	Victory by forfeit or absence of competitor
LIVE	In Progress	In Play	Used in very rare cases. It indicates a fixture in progress but the data indicating the half-time or elapsed time are not available

Fixtures with the status `TBD` may indicate an incorrect fixture date or time because the fixture date or time is not yet known or final. Fixtures with this status are checked and updated daily. The same applies to fixtures with the status `PST`, `CANC`.

The fixtures ids are unique and specific to each fixture. In no case an `ID` will change.

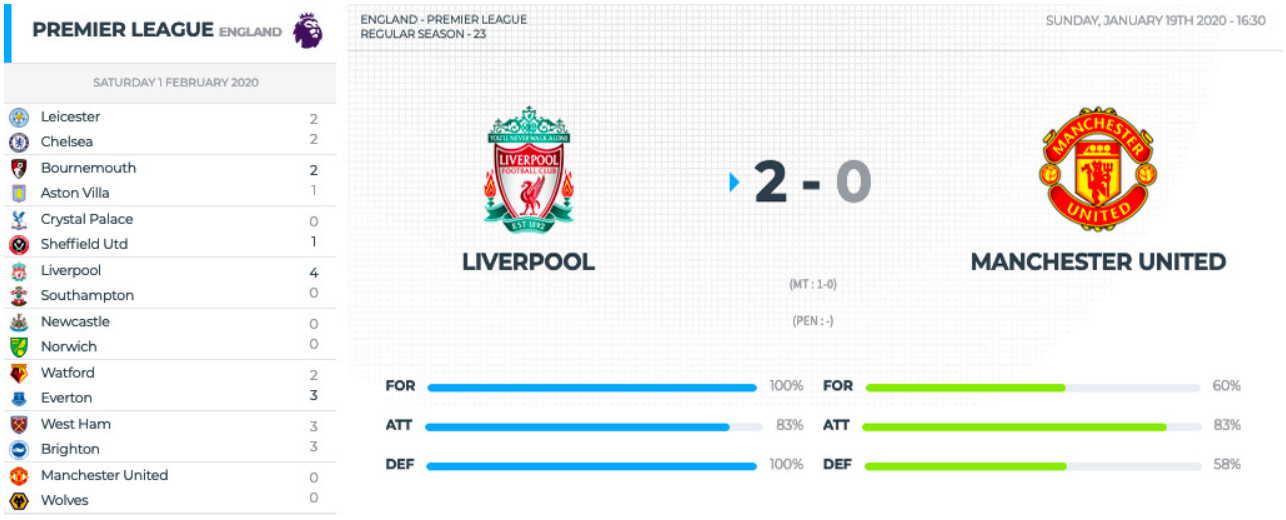
Not all competitions have livescore available and only have `final result`. In this case, the status remains in `NS` and will be updated in the minutes/hours following the match (this can take up to 48 hours, depending on the competition).

> Although the data is updated every 15 seconds, depending on the competition there may be a delay between reality and the availability of data in the API.

Update Frequency : This endpoint is updated every 15 seconds.

Recommended Calls : 1 call per minute for the leagues, teams, fixtures who have at least one fixture in progress otherwise 1 call per day.

> Here are several examples of what can be achieved



QUERY PARAMETERS

`id`	integer

Value: `"id"`

The id of the fixture

`ids`

stringMaximum of 20 fixtures ids

Value: `"id-id-id"`

One or more fixture ids

`live`

string

Enum: `"all"` `"id-id"`

All or several leagues ids

`date`

stringYYYY-MM-DD

A valid date

`league`

integer

The id of the league

`season`

integer = 4 characters YYYY

The season of the league

`team`

integer

The id of the team

`last`

integer <= 2 characters

For the X last fixtures

`next`

integer <= 2 characters

For the X next fixtures

`from`

stringYYYY-MM-DD

A valid date

`to`

stringYYYY-MM-DD

A valid date

`round`

string

The round of the fixture

`status`

string

Enum: `"NS"` `"NS-PST-FT"`

One or more fixture status short

`venue`

integer

The venue id of the fixture

`timezone`

string

A valid timezone from the endpoint `Timezone`

HEADER PARAMETERS

`x-apisports-key` required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

`GET /fixtures`

Request samples

Use Cases | Php | **Python** | Node | JavaScript | Curl | Ruby

Copy

```python
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/fixtures?live=all", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | **204** | **499** | **500**

Content type
application/json

Copy Expand all Collapse all


```
{
   "get": "fixtures",
 - "parameters": {
       "live": "all"
   },
   "errors": [ ],
   "results": 4,
 - "paging": {
       "current": 1,
       "total": 1
   },
 - "response": [
     + { … }
   ]
}
```

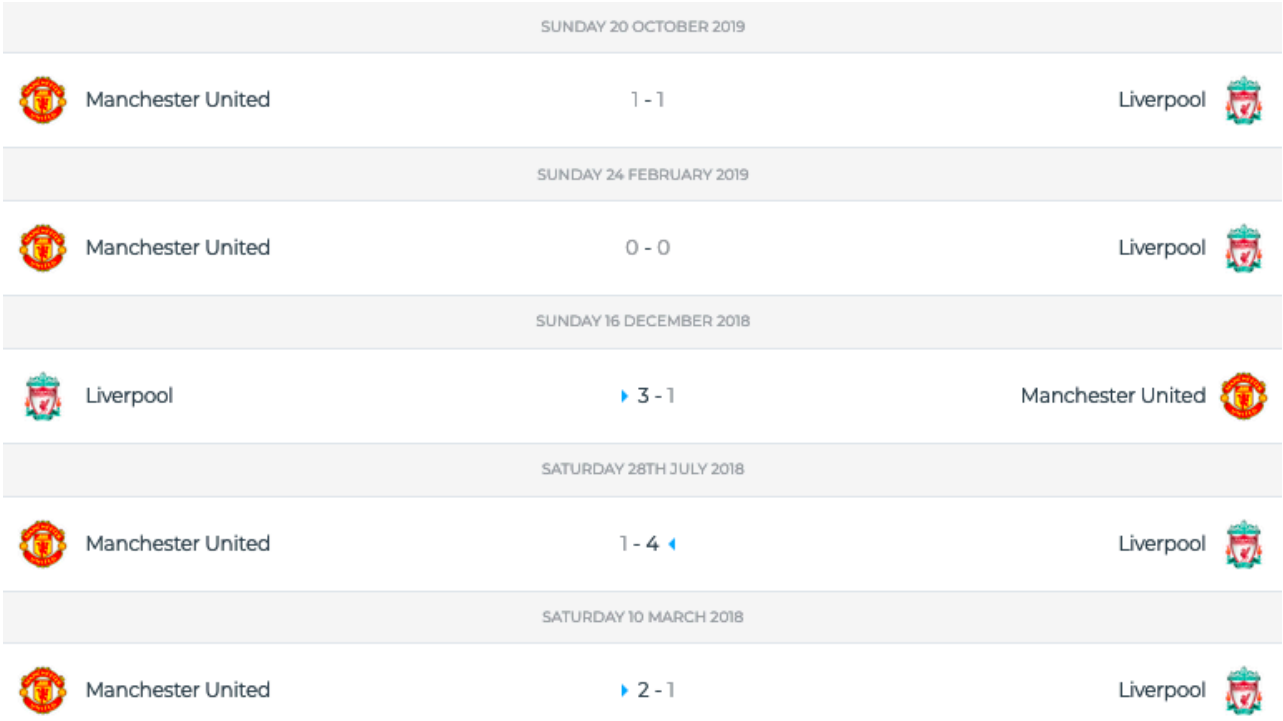
Head To Head

Get heads to heads between two teams.

Update Frequency : This endpoint is updated every 15 seconds.

Recommended Calls : 1 call per minute for the leagues, teams, fixtures who have at least one fixture in progress otherwise 1 call per day.

> Here is an example of what can be achieved



QUERY PARAMETERS

`h2h` *required*
string ID-ID
The ids of the teams

`date`
string YYYY-MM-DD

`league`
integer
The id of the league

`season`
integer = 4 characters YYYY
The season of the league

`last`
integer
For the X last fixtures

`next`
integer
For the X next fixtures

`from`
string YYYY-MM-DD

`to`
string YYYY-MM-DD

`status`
string
Enum: `"NS"` `"NS-PST-FT"`
One or more fixture status short

`venue`
integer
The venue id of the fixture

`timezone`	string A valid timezone from the endpoint `Timezone`

HEADER PARAMETERS

`x-apisports-key` required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

`GET /fixtures/headtohead`

Request samples

Use Cases | Php | **Python** | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/fixtures/headtohead?h2h=33-34", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
  "get": "fixtures/headtohead",
- "parameters": {
    "h2h": "33-34",
    "last": "1"
  },
  "errors": [ ],
  "results": 1,
- "paging": {
    "current": 1,
    "total": 1
  },
- "response": [
   + { ... }
  ]
}
```

Statistics

Get the statistics for one fixture.

Available statistics

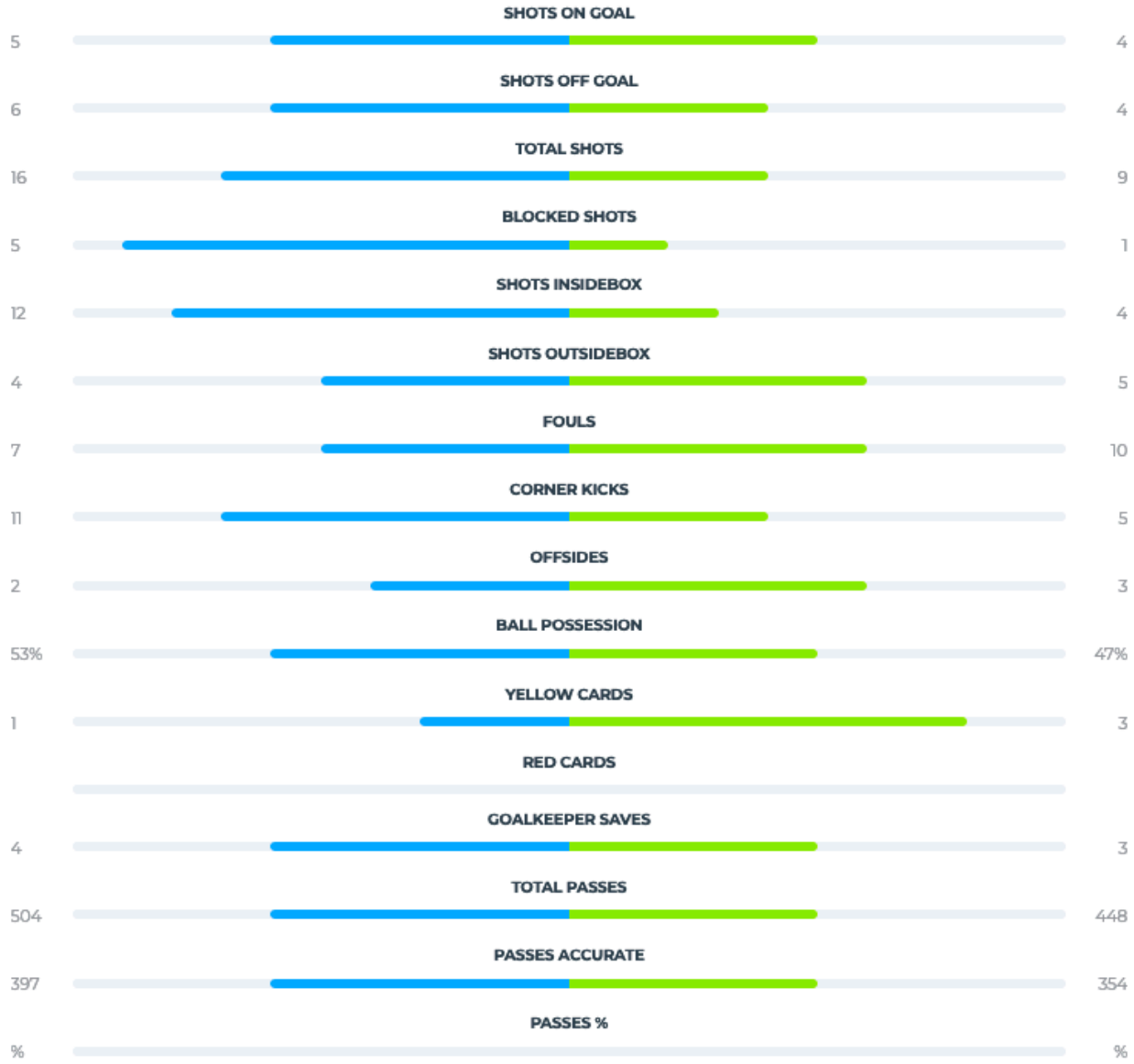
- Shots on Goal
- Shots off Goal
- Shots insidebox
- Shots outsidebox
- Total Shots
- Blocked Shots
- Fouls
- Corner Kicks
- Offsides
- Ball Possession
- Yellow Cards
- Red Cards
- Goalkeeper Saves

- Total passes
- Passes accurate
- Passes %

Update Frequency : This endpoint is updated every minute.

Recommended Calls : 1 call every minute for the teams or fixtures who have at least one fixture in progress otherwise 1 call per day.

> Here is an example of what can be achieved



QUERY PARAMETERS

`fixture` *required*
integer
The id of the fixture

`team`
integer
The id of the team

`type`	string The type of statistics
`half`	boolean Default: `false` Enum: `"true"` `"false"` Add the halftime statistics in the response `Data start from 2024 season for half parameter`

HEADER PARAMETERS

`x-apisports-key` required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /fixtures/statistics

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/fixtures/statistics?fixture=215662&team=463", headers=h
```

```
res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Example
Default

Copy Expand all Collapse all


```
{
    "get": "fixtures/statistics",
  - "parameters": {
        "team": "463",
        "fixture": "215662"
    },
    "errors": [ ],
    "results": 1,
  - "paging": {
        "current": 1,
        "total": 1
    },
  - "response": [
      + { … }
    ]
}
```

Events

Get the events from a fixture.

Available events

TYPE				
Goal	Normal Goal	Own Goal	Penalty	Missed Penalty
Card	Yellow Card	Red card		
Subst	Substitution [1, 2, 3...]			
Var	Goal cancelled	Penalty confirmed		

- *VAR events are available from the 2020-2021 season.*

Update Frequency : This endpoint is updated every 15 seconds.

Recommended Calls : 1 call per minute for the fixtures in progress otherwise 1 call per day.

You can also retrieve all the events of the fixtures in progress with to the endpoint `fixtures?live=all`

> Here is an example of what can be achieved

QUERY PARAMETERS

`fixture` required	integer The id of the fixture
`team`	integer The id of the team
`player`	integer The id of the player
`type`	string The type

HEADER PARAMETERS

`x-apisports-key` required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /fixtures/events

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```python
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/fixtures/events?fixture=215662", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
   "get": "fixtures/events",
```

```
- "parameters": {
      "fixture": "215662"
  },
  "errors": [ ],
  "results": 18,
- "paging": {
      "current": 1,
      "total": 1
  },
- "response": [
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … }
  ]
}
```

Lineups

Get the lineups for a fixture.

Lineups are available between 20 and 40 minutes before the fixture when the competition covers this feature. You can check this with the endpoint `leagues` and the `coverage` field.

> It's possible that for some competitions the lineups are not available before the fixture, in this case, they are updated and available after the match with a variable delay depending on the competition.

Available datas

- Formation
- Coach
- Start XI
- Substitutes

Players' positions on the grid `*`

X = row and **Y** = column (X:Y)

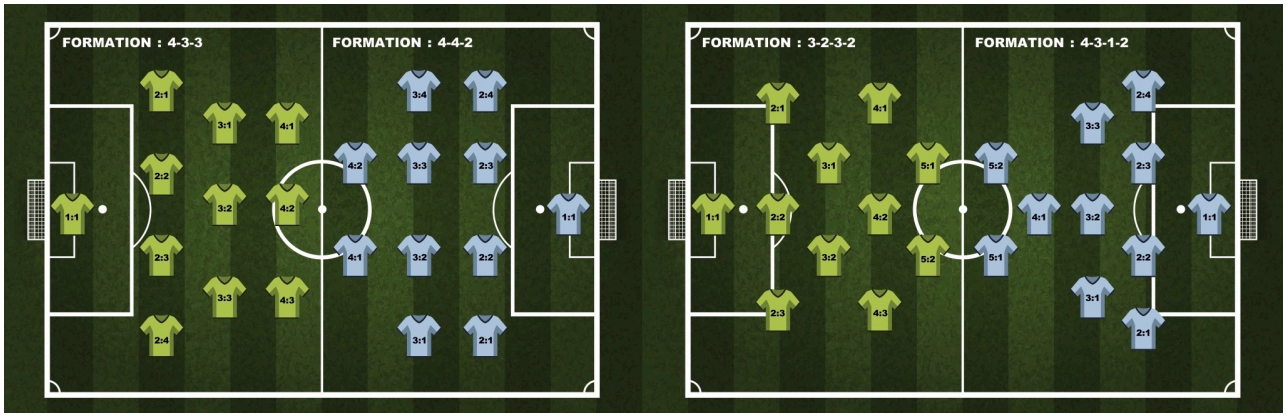
Line 1 **X** being the one of the goal and then for each line this number is incremented. The column **Y** will go from left to right, and incremented for each player of the line.

`* As a new feature, some irregularities may occur, do not hesitate to report them on our public Roadmap`

Update Frequency : This endpoint is updated every 15 minutes.

Recommended Calls : 1 call every 15 minutes for the fixtures in progress otherwise 1 call per day.

> Here are several examples of what can be done



LIVERPOOL 4-3-3	
COACH	
	J. Klopp
STARTING XI	
1	Alisson
4	V. van Dijk
26	A. Robertson
12	J. Gomez
66	T. Alexander-Arnold
5	G. Wijnaldum
14	J. Henderson
15	A. Oxlade-Chamberlain
9	Roberto Firmino
11	Mohamed Salah
10	S. Mané
SUBSTITUTES	
20	A. Lallana
3	Fabinho
27	D. Origi
18	T. Minamino
13	Adrián
48	C. Jones
32	J. Matip

MANCHESTER UNITED 3-4-1-2	
COACH	
	O. Solskjær
STARTING XI	
1	David de Gea
2	V. Lindelöf
5	H. Maguire
23	L. Shaw
29	A. Wan-Bissaka
53	B. Williams
31	N. Matić
17	Fred
15	Andreas Pereira
21	D. James
9	A. Martial
SUBSTITUTES	
8	Mata
26	M. Greenwood
20	Diogo Dalot
22	S. Romero
14	J. Lingard
3	E. Bailly
4	P. Jones

QUERY PARAMETERS

`fixture` *required*
integer
The id of the fixture

`team`
integer
The id of the team

`player`
integer
The id of the player

`type`
string
The type

HEADER PARAMETERS

`x-apisports-key` *required*
string
Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /fixtures/lineups

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/fixtures/lineups?fixture=592872", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
   "get": "fixtures/lineups",
```

```
  - "parameters": {
        "fixture": "592872"
    },
    "errors": [ ],
    "results": 2,
  - "paging": {
        "current": 1,
        "total": 1
    },
  - "response": [
      + { … },
      + { … }
    ]
}
```

Players statistics

Get the players statistics from one fixture.

Update Frequency : This endpoint is updated every minute.

Recommended Calls : 1 call every minute for the fixtures in progress otherwise 1 call per day.

QUERY PARAMETERS

`fixture` required	integer The id of the fixture
`team`	integer The id of the team

HEADER PARAMETERS

`x-apisports-key` required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

```
GET  /fixtures/players
```

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```python
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/fixtures/players?fixture=169080", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
   "get": "fixtures/players",
 - "parameters": {
       "fixture": "169080"
   },
   "errors": [ ],
```

```
    "results": 2,
  - "paging": {
        "current": 1,
        "total": 1
    },
  - "response": [
     + { … }
    ]
}
```


Injuries

Injuries

Get the list of players not participating in the fixtures for various reasons such as `suspended`, `injured` for example.

Being a new endpoint, the data is only available from April 2021.

There are two types:

- `Missing Fixture` : The player will not play the fixture.
- `Questionable` : The information is not yet 100% sure, the player may eventually play the fixture.

> Examples available in Request samples "Use Cases".

> All the parameters of this endpoint can be used together.

This endpoint requires at least one parameter.

Update Frequency : This endpoint is updated every 4 hours.

Recommended Calls : 1 call per day.

QUERY PARAMETERS

`league`	integer The id of the league
`season`	integer `= 4 characters` YYYY

	The season of the league, **required** with `league`, `team` and `player` parameters
`fixture`	integer The id of the fixture
`team`	integer The id of the team
`player`	integer The id of the player
`date`	stringYYYY-MM-DD A valid date
`ids`	stringMaximum of 20 fixtures ids Value: `"id-id-id"` One or more fixture ids
`timezone`	string A valid timezone from the endpoint `Timezone`

HEADER PARAMETERS

`x-apisports-key` `required`	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET `/injuries`

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```python
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/injuries?fixture=686314", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "injuries",
  - "parameters": {
        "fixture": "686314"
    },
    "errors": [ ],
    "results": 13,
  - "paging": {
        "current": 1,
        "total": 1
    },
```

```
- "response": [
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... }
  ]
}
```

Predictions

Predictions

Get predictions about a fixture.

The predictions are made using several algorithms including the poisson distribution, comparison of team statistics, last matches, players etc...

Bookmakers odds are not used to make these predictions

Also provides some comparative statistics between teams

Available Predictions

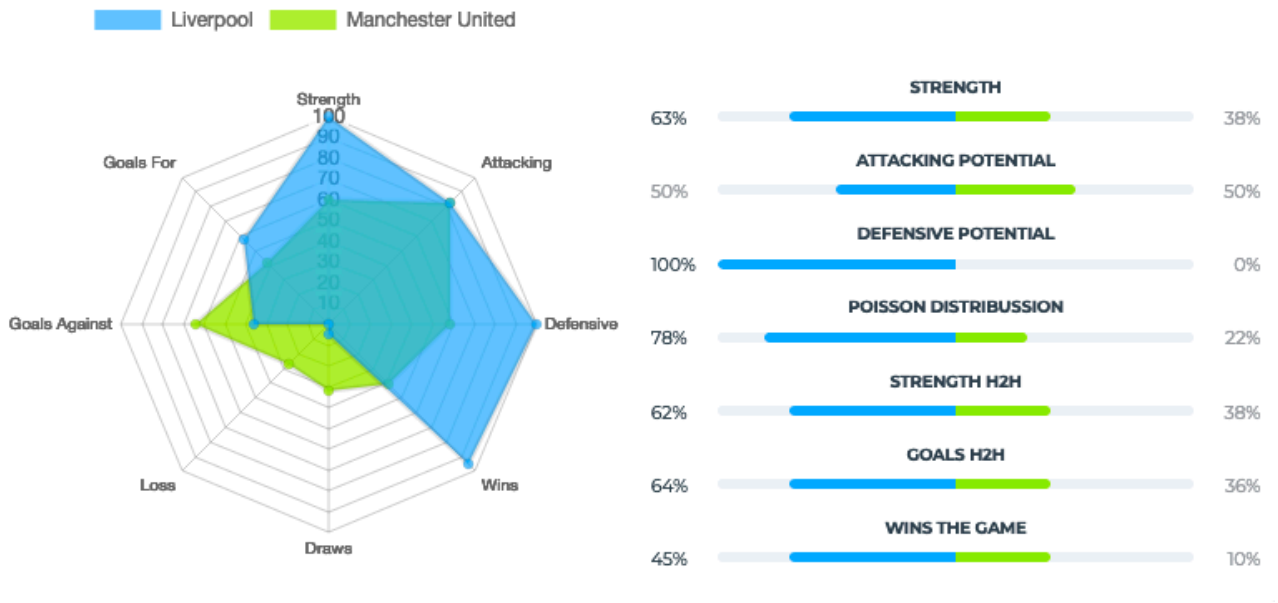
- Match winner : Id of the team that can potentially win the fixture
- Win or Draw : If `True` indicates that the designated team can win or draw
- Under / Over : -1.5 / -2.5 / -3.5 / -4.5 / +1.5 / +2.5 / +3.5 / +4.5 `*`
- Goals Home : -1.5 / -2.5 / -3.5 / -4.5 `*`
- Goals Away -1.5 / -2.5 / -3.5 / -4.5 `*`
- Advice *(Ex : Deportivo Santani or draws and -3.5 goals)*

`*` **-1.5** means that there will be a maximum of **1.5** goals in the fixture, i.e : **1** goal

Update Frequency : This endpoint is updated every hour.

Recommended Calls : 1 call per hour for the fixtures in progress otherwise 1 call per day.

> Here is an example of what can be achieved



QUERY PARAMETERS

`fixture`
required

integer
The id of the fixture

HEADER PARAMETERS

`x-apisports-key`
required

string
Your Api-Key

Responses

- **200** OK
- **204** No Content
- **499** Time Out
- **500** Internal Server Error

GET /predictions

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/predictions?fixture=198772", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
   "get": "predictions",
 - "parameters": {
       "fixture": "198772"
   },
   "errors": [ ],
   "results": 1,
 - "paging": {
       "current": 1,
       "total": 1
   },
 - "response": [
     + { ... }
   ]
```

```
}
```

Coachs

Coachs

Get all the information about the coachs and their careers.

To get the photo of a coach you have to call the following url: `https://media.api-sports.io/football/coachs/{coach_id}.png`

Update Frequency : This endpoint is updated every day.

Recommended Calls : 1 call per day.

QUERY PARAMETERS

`id`	integer The id of the coach
`team`	integer The id of the team
`search`	string >= 3 characters The name of the coach

HEADER PARAMETERS

`x-apisports-key` required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

```
GET  /coachs
```

Request samples

Use Cases | Php | **Python** | Node | JavaScript | Curl | Ruby

Copy

```python
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/coachs?team=85", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | **204** | **499** | **500**

Content type
application/json

Copy Expand all Collapse all


```
{
   "get": "coachs",
 - "parameters": {
       "team": "85"
   },
   "errors": [ ],
   "results": 1,
```

```
  - "paging": {
        "current": 1,
        "total": 1
    },
  - "response": [
      + { ... }
    ]
}
```


Players

Seasons

Get all available seasons for players statistics.

Update Frequency : This endpoint is updated every day.

Recommended Calls : 1 call per day.

QUERY PARAMETERS

`player`	integer The id of the player

HEADER PARAMETERS

`x-apisports-key` required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /players/seasons

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```python
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/players/seasons", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "players/seasons",
    "parameters": [ ],
    "errors": [ ],
    "results": 35,
```

```
    - "paging": {
        "current": 1,
        "total": 1
      },
    - "response": [
        1966,
        1982,
        1986,
        1990,
        1991,
        1992,
        1993,
        1994,
        1995,
        1996,
        1997,
        1998,
        1999,
        2000,
        2001,
        2002,
        2003,
        2004,
        2005,
        2006,
        2007,
        2008,
        2009,
        2010,
        2011,
        2012,
        2013,
        2014,
        2015,
        2016,
        2017,
        2018,
        2019,
        2020,
        2022
      ]
}
```

Profiles

Returns the list of all available players.

It is possible to call this endpoint without parameters, but you will need to use the **pagination** to get all available players.

To get the photo of a player you have to call the following url: `https://media.api-sports.io/football/players/{player_id}.png`

This endpoint uses a **pagination system**, you can navigate between the different pages with to the `page` parameter.

> **Pagination** : 250 results per page.

Update Frequency : This endpoint is updated several times a week.

Recommended Calls : 1 call per week.

QUERY PARAMETERS

`player`	integer The id of the player
`search`	string `>= 3 characters` The lastname of the player
`page`	integer Default: `1` Use for the pagination

HEADER PARAMETERS

`x-apisports-key` `required`	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

`GET /players/profiles`

Request samples

Use Cases | Php | **Python** | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/players/profiles?player=276", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "players/profiles",
  - "parameters": {
        "player": "276"
    },
    "errors": [ ],
    "results": 1,
```

```
- "paging": {
      "current": 1,
      "total": 1
  },
- "response": [
    + { ... }
  ]
}
```

Statistics

Get players statistics.

This endpoint returns the players for whom the `profile` and `statistics` data are available. Note that it is possible that a player has statistics for 2 teams in the same season in case of transfers.

The statistics are calculated according to the team `id`, league `id` and `season`.

You can find the available `seasons` by using the endpoint `players/seasons`.

> To get the squads of the teams it is better to use the endpoint `players/squads`.

The players `id` are unique in the API and players keep it among all the teams they have been in.

In this endpoint you have the `rating` field, which is the rating of the player according to a match or a season. This data is calculated according to the performance of the player in relation to the other players of the game or the season who occupy the same position *(Attacker, defender, goal...)*. There are different algorithms that take into account the position of the player and assign points according to his performance.

To get the photo of a player you have to call the following url: `https://media.api-sports.io/football/players/{player_id}.png`

This endpoint uses a **pagination system**, you can navigate between the different pages with to the `page` parameter.

> **Pagination** : 20 results per page.

Update Frequency : This endpoint is updated several times a week.

Recommended Calls : 1 call per day.

Tutorials :

- HOW TO GET ALL TEAMS AND PLAYERS FROM A LEAGUE ID

QUERY PARAMETERS

id	integer The id of the player
team	integer The id of the team
league	integer The id of the league
season	integer = 4 characters YYYY | Requires the fields Id, League or Team... The season of the league
search	string >= 4 characters Requires the fields League or Team The name of the player
page	integer Default: 1 Use for the pagination

HEADER PARAMETERS

x-apisports-key required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /players

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```python
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/players?id=276&season=2019", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
   "get": "players",
 - "parameters": {
       "id": "276",
       "season": "2019"
   },
   "errors": [ ],
   "results": 1,
 - "paging": {
       "current": 1,
       "total": 1
   },
 - "response": [
     + { … }
   ]
}
```

Squads

Return the current squad of a team when the `team` parameter is used. When the `player` parameter is used the endpoint returns the set of teams associated with the player.

> The response format is the same regardless of the parameter sent.

This endpoint requires at least one parameter.

Update Frequency : This endpoint is updated several times a week.

Recommended Calls : 1 call per week.

QUERY PARAMETERS

`team`	integer The id of the team
`player`	integer The id of the player

HEADER PARAMETERS

`x-apisports-key` `required`	string Your Api-Key

Responses

- **200** OK
- **204** No Content
- **499** Time Out
- **500** Internal Server Error

`GET /players/squads`

Request samples

Use Cases | Php | **Python** | Node | JavaScript | Curl | Ruby

Copy

```python
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/players/squads?team=33", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
   "get": "players/squads",
 - "parameters": {
       "team": "33"
   },
   "errors": [ ],
   "results": 1,
 - "paging": {
       "current": 1,
       "total": 1
   },
 - "response": [
     + { ... }
   ]
}
```

Teams

Returns the list of teams and seasons in which the player played during his career.

This endpoint requires at least one parameter.

Update Frequency : This endpoint is updated several times a week.

Recommended Calls : 1 call per week.

QUERY PARAMETERS

`player` required	integer The id of the player

HEADER PARAMETERS

`x-apisports-key` required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

```
GET  /players/teams
```

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```
import http.client
```

```
conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/players/teams?player=276", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
   "get": "players/teams",
 - "parameters": {
       "player": "276"
   },
   "errors": [ ],
   "results": 8,
 - "paging": {
       "current": 1,
       "total": 1
   },
 - "response": [
     + { … },
     + { … },
     + { … },
     + { … },
     + { … },
     + { … },
     + { … },
     + { … }
   ]
}
```

Top Scorers

Get the 20 best players for a league or cup.

How it is calculated:

- 1 : The player that has scored the higher number of goals
- 2 : The player that has scored the fewer number of penalties
- 3 : The player that has delivered the higher number of goal assists
- 4 : The player that scored their goals in the higher number of matches
- 5 : The player that played the fewer minutes
- 6 : The player that plays for the team placed higher on the table
- 7 : The player that received the fewer number of red cards
- 8 : The player that received the fewer number of yellow cards

Update Frequency : This endpoint is updated several times a week.

Recommended Calls : 1 call per day.

QUERY PARAMETERS

`league` required	integer The id of the league
`season` required	integer = 4 characters YYYY The season of the league

HEADER PARAMETERS

`x-apisports-key` required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /players/topscorers

Request samples

Php | Python | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/players/topscorers?season=2018&league=61", headers=head

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
   "get": "players/topscorers",
 - "parameters": {
       "league": "61",
       "season": "2018"
   },
   "errors": [ ],
   "results": 20,
 - "paging": {
       "current": 1,
       "total": 1
   },
```

```
- "response": [
    + { ... },
    + { ... }
  ]
}
```


Top Assists

Get the 20 best players assists for a league or cup.

How it is calculated:

- 1 : The player that has delivered the higher number of goal assists
- 2 : The player that has scored the higher number of goals
- 3 : The player that has scored the fewer number of penalties
- 4 : The player that assists in the higher number of matches
- 5 : The player that played the fewer minutes
- 6 : The player that received the fewer number of red cards
- 7 : The player that received the fewer number of yellow cards

Update Frequency : This endpoint is updated several times a week.

Recommended Calls : 1 call per day.

QUERY PARAMETERS

`league` required	integer The id of the league
`season` required	integer = 4 characters YYYY The season of the league

HEADER PARAMETERS

`x-apisports-key` required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

```
GET  /players/topassists
```

Request samples

Php | Python | Node | JavaScript | Curl | Ruby

Copy

```python
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/players/topassists?season=2020&league=61", headers=head

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "players/topassists",
  - "parameters": {
        "season": "2020",
        "league": "61"
    },
    "errors": [ ],
```

```
    "results": 20,
  - "paging": {
        "current": 0,
        "total": 1
    },
  - "response": [
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … }
    ]
}
```

Top Yellow Cards

Get the 20 players with the most yellow cards for a league or cup.

How it is calculated:

- 1 : The player that received the higher number of yellow cards
- 2 : The player that received the higher number of red cards
- 3 : The player that assists in the higher number of matches
- 4 : The player that played the fewer minutes

Update Frequency : This endpoint is updated several times a week.

Recommended Calls : 1 call per day.

QUERY PARAMETERS

league required	integer The id of the league
season required	integer = 4 characters YYYY The season of the league

HEADER PARAMETERS

x-apisports-key required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /players/topyellowcards

Request samples

Php | Python | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/players/topyellowcards?season=2020&league=61", headers=
```

```
res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "players/topyellowcards",
  - "parameters": {
        "season": "2020",
        "league": "61"
    },
    "errors": [ ],
    "results": 20,
  - "paging": {
        "current": 0,
        "total": 1
    },
```

```
- "response": [
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... }
  ]
}
```

Top Red Cards

Get the 20 players with the most red cards for a league or cup.

How it is calculated:

- 1 : The player that received the higher number of red cards
- 2 : The player that received the higher number of yellow cards
- 3 : The player that assists in the higher number of matches
- 4 : The player that played the fewer minutes

Update Frequency : This endpoint is updated several times a week.

Recommended Calls : 1 call per day.

QUERY PARAMETERS

`league` required	integer The id of the league

season required	integer = 4 characters YYYY The season of the league

HEADER PARAMETERS

x-apisports-key required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /players/topredcards

Request samples

Php | Python | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/players/topredcards?season=2020&league=61", headers=hea

res = conn.getresponse()
data = res.read()
```

```
print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
  "get": "players/topredcards",
- "parameters": {
    "season": "2020",
    "league": "61"
  },
  "errors": [ ],
  "results": 20,
- "paging": {
    "current": 0,
    "total": 1
  },
- "response": [
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … }
  ]
```

```
}
```

Transfers

Transfers

Get all available transfers for players and teams

Update Frequency : This endpoint is updated several times a week.

Recommended Calls : 1 call per day.

QUERY PARAMETERS

`player`	integer The id of the player
`team`	integer The id of the team

HEADER PARAMETERS

`x-apisports-key` required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /transfers

Request samples

Use Cases | Php | **Python** | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/transfers?player=35845", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
  "get": "transfers",
- "parameters": {
    "player": "35845"
  },
  "errors": [ ],
  "results": 1,
```

```
  - "paging": {
        "current": 1,
        "total": 1
    },
  - "response": [
      + { ... }
    ]
}
```


Trophies

Trophies

Get all available trophies for a player or a coach.

Update Frequency : This endpoint is updated several times a week.

Recommended Calls : 1 call per day.

QUERY PARAMETERS

`player` integer
The id of the player

`players` stringMaximum of 20 players ids
Value: `"id-id-id"`
One or more players ids

`coach` integer
The id of the coach

`coachs` stringMaximum of 20 coachs ids
Value: `"id-id-id"`
One or more coachs ids

HEADER PARAMETERS

x-apisports-key required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /trophies

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/trophies?player=276", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Example
Default

Copy Expand all Collapse all


```
{
   "get": "trophies",
 - "parameters": {
       "player": "276"
   },
   "errors": [ ],
   "results": 38,
 - "paging": {
       "current": 1,
       "total": 1
   },
 - "response": [
     + { … },
     + { … },
     + { … },
     + { … },
     + { … },
     + { … },
     + { … },
     + { … },
     + { … },
     + { … }
   ]
}
```

Sidelined

Sidelined

Get all available sidelined for a player or a coach.

Update Frequency : This endpoint is updated several times a week.

Recommended Calls : 1 call per day.

QUERY PARAMETERS

`player`	integer The id of the player
`players`	stringMaximum of 20 players ids Value: `"id-id-id"` One or more players ids
`coach`	integer The id of the coach
`coachs`	stringMaximum of 20 coachs ids Value: `"id-id-id"` One or more coachs ids

HEADER PARAMETERS

`x-apisports-key` required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /sidelined

Request samples

Use Cases | Php | **Python** | Node | JavaScript | Curl | Ruby

Copy

```python
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/sidelined?player=276", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Example
Default

Copy Expand all Collapse all


```
{
    "get": "sidelined",
  - "parameters": {
        "player": "276"
    },
    "errors": [ ],
    "results": 27,
  - "paging": {
        "current": 1,
        "total": 1
    },
```

```
- "response": [
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... }
  ]
}
```

Odds (In-Play)

odds/live

This endpoint returns in-play odds for fixtures in progress.

Fixtures are added between 15 and 5 minutes before the start of the fixture. Once the fixture is over they are removed from the endpoint between 5 and 20 minutes. **No history is stored**. So fixtures that are about to start, fixtures in progress and fixtures that have just ended are available in this endpoint.

Update Frequency : This endpoint is updated every 5 seconds. `*`

`* This value can change in the range of 5 to 60 seconds`

INFORMATIONS ABOUT STATUS

```
"status": {
    "stopped": false, // True if the fixture is stopped by the referee for X re
    "blocked": false, // True if bets on this fixture are temporarily blocked
    "finished": false // True if the fixture has not started or if it is finish
},
```

INFORMATIONS ABOUT VALUES

When several identical values exist for the same bet the `main` field is set to `True` for the bet being considered, the others will have the value `False` .

The `main` field will be set to `True` only if several identical values exist for the same bet.

When a value is unique for a bet the `main` value will always be `False` or `null` .

Example below :

```
"id": 36,
"name": "Over/Under Line",
"values": [
    {
        "value": "Over",
        "odd": "1.975",
        "handicap": "2",
        "main": true, // Bet to consider
        "suspended": false // True if this bet is temporarily suspended
    },
    {
        "value": "Over",
        "odd": "3.45",
        "handicap": "2",
        "main": false, // Bet to no consider
        "suspended": false
    },
]
```

QUERY PARAMETERS

`fixture`	integer The id of the fixture

league	integer (In this endpoint the "season" parameter is ... Show pattern The id of the league
bet	integer The id of the bet

HEADER PARAMETERS

x-apisports-key required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /odds/live

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/odds/live?bet=1&league=39", headers=headers)

res = conn.getresponse()
data = res.read()
```

```
print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "odds/live",
  - "parameters": {
        "fixture": "721238"
    },
    "errors": [ ],
    "results": 1,
  - "paging": {
        "current": 1,
        "total": 1
    },
  - "response": [
      + { ... }
    ]
}
```


---

odds/live/bets

Get all available bets for in-play odds.

All bets `id` can be used in endpoint `odds/live` as filters, **but are not compatible with endpoint `odds` for pre-match odds**.

Update Frequency : This endpoint is updated every 60 seconds.

QUERY PARAMETERS

id	string The id of the bet name

search
string = 3 characters
The name of the bet

HEADER PARAMETERS

x-apisports-key
required
string
Your Api-Key

Responses

> 200 OK

> 204 No Content

> 499 Time Out

> 500 Internal Server Error

GET /odds/live/bets

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/odds/bets", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
  "get": "odds/live/bets",
  "parameters": [ ],
  "errors": [ ],
  "results": 137,
- "paging": {
    "current": 1,
    "total": 1
  },
```

```
- "response": [
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• },
    + { ••• }
```

+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },
+ { ••• },

```
        + { ... }
      ]
    }
```


Odds (Pre-Match)

Odds

Get odds from fixtures, leagues or date.

This endpoint uses a **pagination system**, you can navigate between the different pages with to the `page` parameter.

> **Pagination** : 10 results per page.

We provide pre-match odds between 1 and 14 days before the fixture.

We keep a 7-days history *(The availability of odds may vary according to the leagues, seasons, fixtures and bookmakers)*

Update Frequency : This endpoint is updated every 3 hours.

Recommended Calls : 1 call every 3 hours.

QUERY PARAMETERS

`fixture`	integer The id of the fixture
`league`	integer The id of the league
`season`	integer = 4 characters YYYY The season of the league
`date`	stringYYYY-MM-DD A valid date
`timezone`	string A valid timezone from the endpoint `Timezone`

page	integer Default: 1 Use for the pagination
bookmaker	integer The id of the bookmaker
bet	integer The id of the bet

HEADER PARAMETERS

x-apisports-key required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /odds

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }
```

```
conn.request("GET", "/odds?season=2019&bet=1&bookmaker=6&fixture=157140&leagu

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "odds",
  - "parameters": {
        "fixture": "326090",
        "bookmaker": "6"
    },
    "errors": [ ],
    "results": 1,
  - "paging": {
        "current": 1,
        "total": 1
    },
  - "response": [
      + { … }
    ]
}
```

Mapping

Get the list of available fixtures `id` for the endpoint odds.

All fixtures, leagues `id` and `date` can be used in endpoint odds as filters.

This endpoint uses a **pagination system**, you can navigate between the different pages with to the `page` parameter.

> **Pagination** : 100 results per page.

Update Frequency : This endpoint is updated every day.

Recommended Calls : 1 call per day.

QUERY PARAMETERS

`page`	integer Default: `1` Use for the pagination

HEADER PARAMETERS

`x-apisports-key` required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /odds/mapping

Request samples

Php | Python | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }
```

```
conn.request("GET", "/odds/mapping", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "odds/mapping",
    "parameters": [ ],
    "errors": [ ],
    "results": 129,
  - "paging": {
        "current": 1,
        "total": 1
    },
  - "response": [
      + { … },
      + { … },
      + { … },
      + { … }
    ]
}
```

Bookmakers

Get all available bookmakers.

All bookmakers `id` can be used in endpoint odds as filters.

Update Frequency : This endpoint is updated several times a week.

Recommended Calls : 1 call per day.

QUERY PARAMETERS

id	integer The id of the bookmaker
search	string = 3 characters The name of the bookmaker

HEADER PARAMETERS

x-apisports-key required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /odds/bookmakers

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")

headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/odds/bookmakers", headers=headers)

res = conn.getresponse()
data = res.read()
```

```
print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "odds/bookmakers",
    "parameters": [ ],
    "errors": [ ],
    "results": 15,
  - "paging": {
        "current": 1,
        "total": 1
    },
  - "response": [
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … }
    ]
}
```


Bets

Get all available bets for pre-match odds.

All bets `id` can be used in endpoint odds as filters, **but are not compatible with endpoint `odds/live` for in-play odds**.

Update Frequency : This endpoint is updated several times a week.

Recommended Calls : 1 call per day.

QUERY PARAMETERS

`id`	string The id of the bet name
`search`	string = 3 characters The name of the bet

HEADER PARAMETERS

`x-apisports-key` required	string Your Api-Key

Responses

> **200** OK

> **204** No Content

> **499** Time Out

> **500** Internal Server Error

GET /odds/bets

Request samples

Use Cases | Php | Python | Node | JavaScript | Curl | Ruby

Copy

```
import http.client

conn = http.client.HTTPSConnection("v3.football.api-sports.io")
```

```
headers = {
    'x-apisports-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

conn.request("GET", "/odds/bets", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Response samples

200 | 204 | 499 | 500

Content type
application/json

Copy Expand all Collapse all


```
{
   "get": "odds/bets",
 - "parameters": {
       "search": "under"
   },
   "errors": [ ],
   "results": 7,
 - "paging": {
       "current": 1,
       "total": 1
   },
 - "response": [
     + { … },
     + { … },
     + { … },
     + { … },
     + { … },
     + { … },
     + { … }
   ]
}
```