

# API-BASKETBALL (1.5.2)

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

## Introduction

Welcome to Api-Basketball! You can use our API to access all API endpoints, which can get information about Basketball 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.

## Authentication

We use API keys to allow access to the API. You can register a new API key in [rapidapi](#) or directly on our [dashboard](#).

The accounts on **RapidAPI** and on our **Dashboard** are dissociated. Each of these registration methods has its own **URL** and **API-KEY**. You must therefore adapt your scripts according to your subscription by adapting the URL and your API-KEY.

**RAPIDAPI** : <https://api-basketball.p.rapidapi.com/>

**API-SPORTS** : <https://v1.basketball.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 `XXXXXXXXXXXXXXXXXXXXXXXXXXXX` 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-rapidapi-host`
- `x-rapidapi-key`
- `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.

## RAPIDAPI Account

All information related to your subscription are available on the rapidApi developer dashboard.

The RapidAPI developer dashboard is where you can see all of your apps, locate API keys, view analytics, and manage billing settings.

To access the dashboard, simply login to RapidAPI and select 'My Apps' in the top-right menu. Alternatively, you can head directly to <https://rapidapi.com/developer/dashboard>.

In the main dashboard, you will see account-wide analytics and account information. To get more detailed information, you can select tabs on the left-hand side of the screen.

### App Specific Analytics

Using the RapidAPI dashboard, you can also view analytics specific to each app in your account. To do so, switch over to the 'Analytics' tab of your application in the dashboard.

On the top of the page, you'll be able to see a chart with all the calls being made to all the APIs your app is connected to. You'll also be able to see a log with all the request data. You are also able to filter these analytics to only show certain APIs within the app.

In each graph, you can view the following metrics:

- `API Calls`: how many requests are being made
- `Error rates`: how many requests are error some
- `Latency`: how long (on average) requests take to execute

You may change the time period you're looking at by clicking the calendar icon and choosing a time range.

### Headers sent as response

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

- `server`: The current version of the API proxy used by RapidAPI.
- `x-ratelimit-requests-limit`: The number of requests the plan you are currently subscribed to allows you to make, before incurring overages.

- `x-ratelimit-requests-remaining`: The number of requests remaining before you reach the limit of requests your application is allowed to make, before experiencing overage charges.
- `X-RapidAPI-Proxy-Response`: This header is set to true when the RapidAPI proxy generates the response, (i.e. the response is not generated from our servers)

## 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://v1.basketball.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
    }
  }
}
```

```

}
}

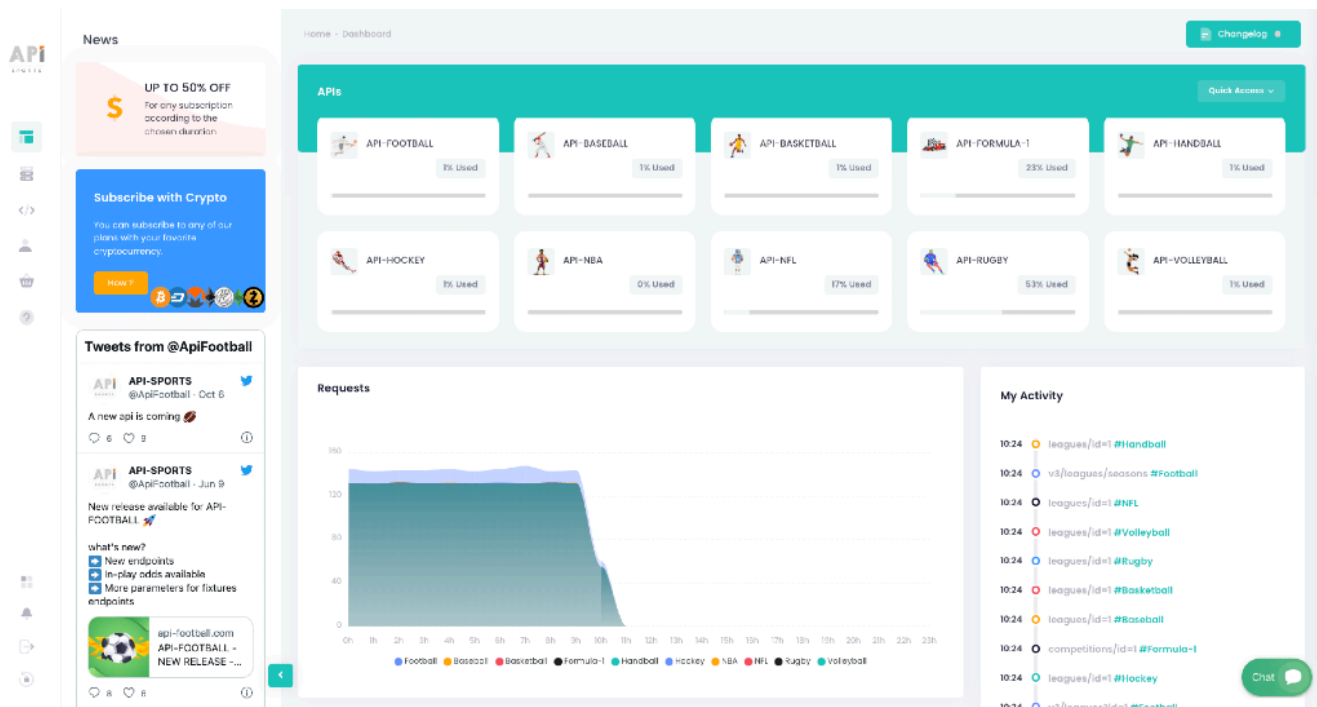
```

## Headers sent as response

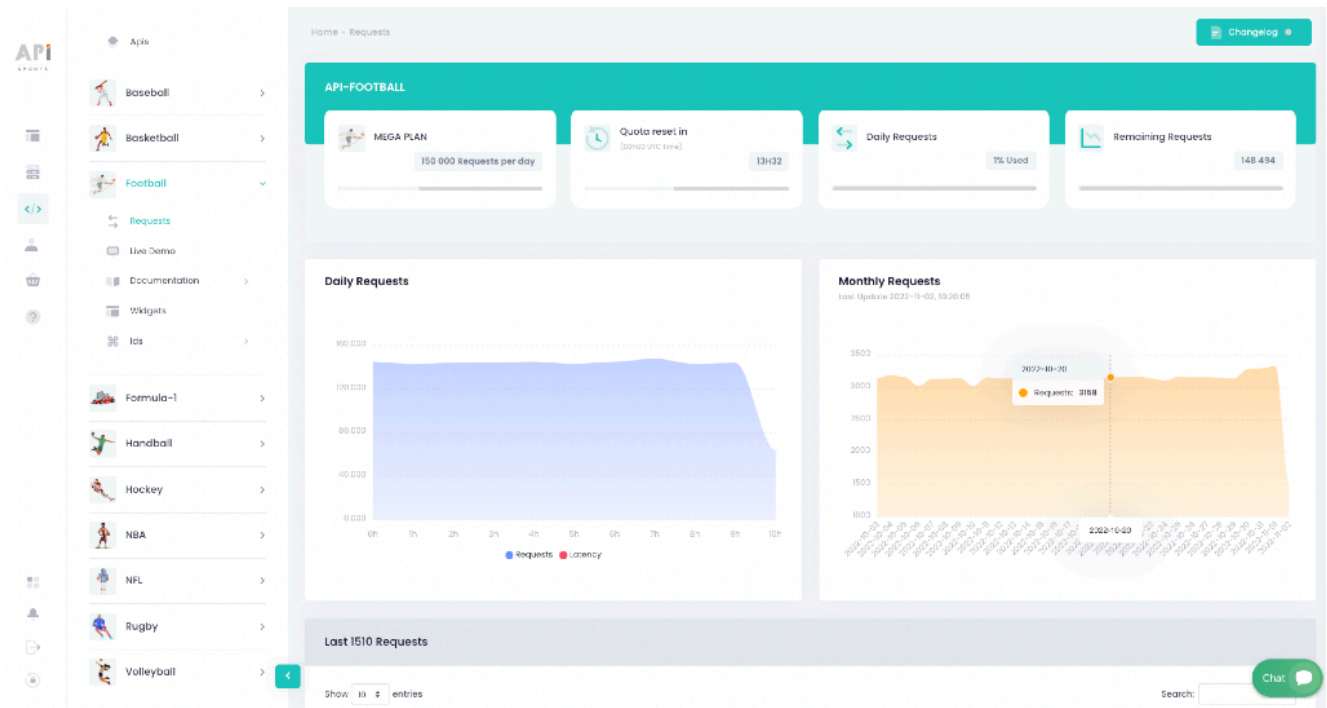
When consuming our API, you will always receive the following headers appended to the response:

- `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.

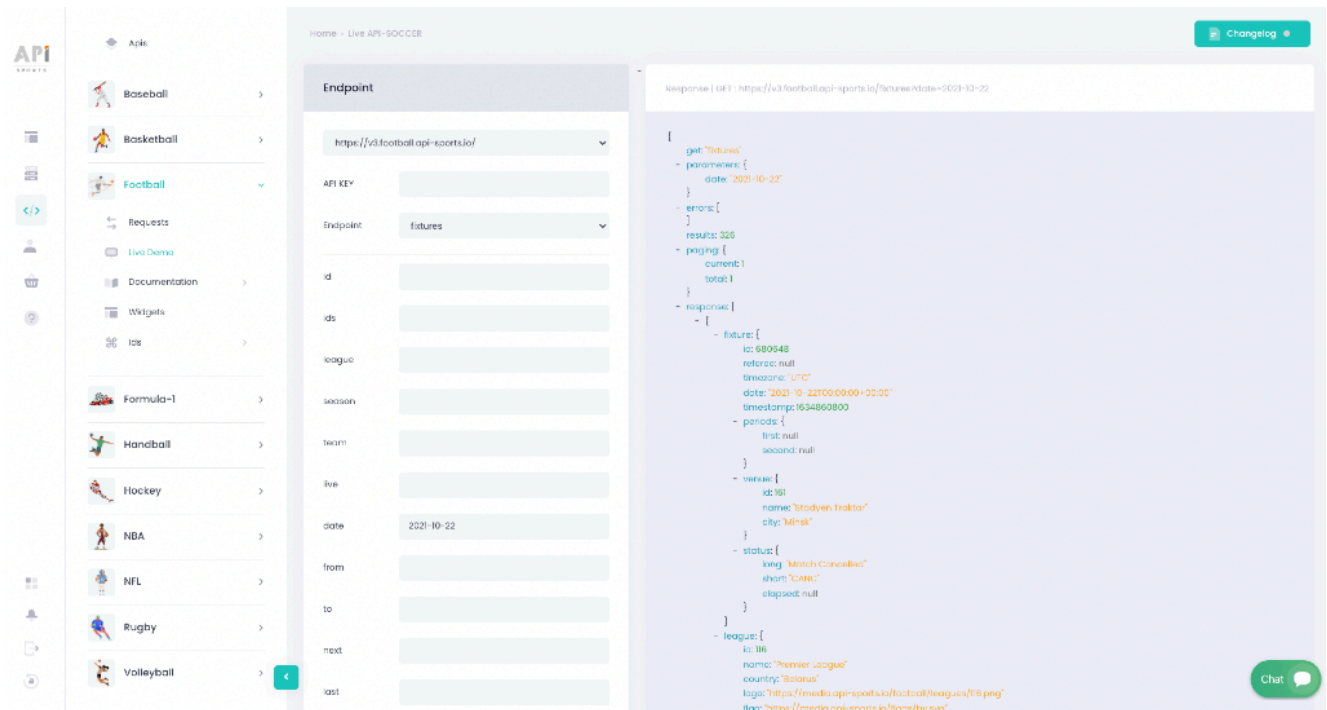
## Dashboard



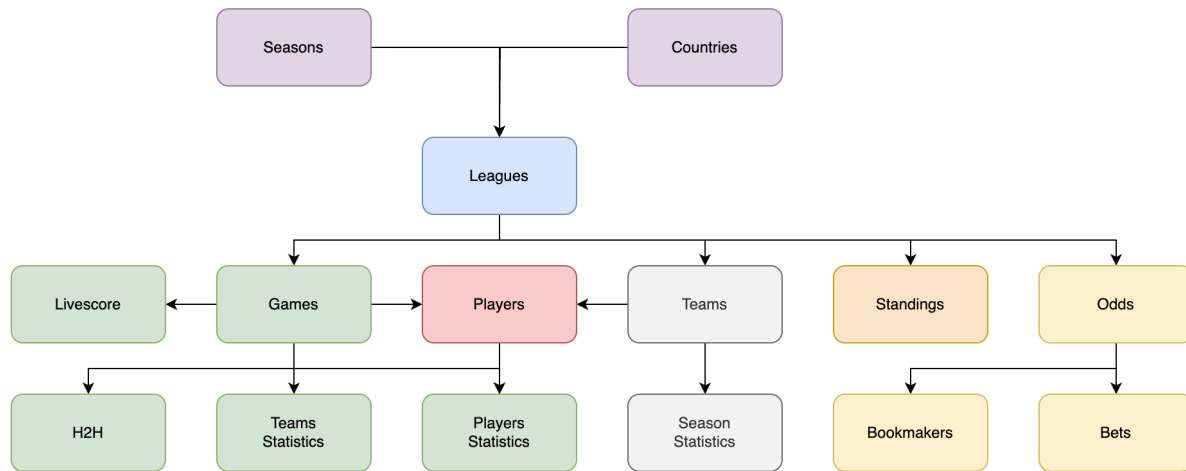
## Requests



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](https://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 `games` for example. In all the sample scripts we will use the `leagues` endpoint as example.

Also you will have to replace `XXXXXXXXXXXXXXXXXXXXXXXXXX` with your API-KEY provided in the [dashboard](#) or on [rapidapi](#).

## 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://v1.basketball.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-rapidapi-key: XXXXXXXXXXXXXXXXXXXX");
    headers = curl_slist_append(headers, "x-rapidapi-host: v1.basketball.api-sports.");
    curl_easy_setopt(curl, CURLOPT_HTTPHEADER, headers);
    res = curl_easy_perform(curl);
}
curl_easy_cleanup(curl);
```

## C#

`RestSharp`

```
var client = new RestClient("https://v1.basketball.api-sports.io/leagues");
client.Timeout = -1;
var request = new RestRequest(Method.GET);
request.AddHeader("x-rapidapi-key", "XXXXXXXXXXXXXXXXXXXXXXXXX");
request.AddHeader("x-rapidapi-host", "v1.basketball.api-sports.io");
```

```
IRestResponse response = client.Execute(request);  
Console.WriteLine(response.Content);
```

## cURL

Curl

```
curl --request GET \  
  --url https://v1.basketball.api-sports.io/leagues \  
  --header 'x-rapidapi-host: v1.basketball.api-sports.io' \  
  --header 'x-rapidapi-key: XXXXXXXXXXXXXXXXXXXX'
```

## Dart

http

```
var headers = {  
  'x-rapidapi-key': 'XXXXXXXXXXXXXXXXXXXXXXXX',  
  'x-rapidapi-host': 'v1.basketball.api-sports.io'  
};  
var request = http.Request('GET', Uri.parse('https://v1.basketball.api-sports.io/1  
  
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://v1.basketball.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-rapidapi-key", "XxXxXxXxXxXxXxXxXxXxXxXxXx")
    req.Header.Add("x-rapidapi-host", "v1.basketball.api-sports.io")

    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-rapidapi-key", "XXXXXXXXXXXXXXXXXXXXXXXXXXXX");  
myHeaders.append("x-rapidapi-host", "v1.basketball.api-sports.io");  
  
var requestOptions = {  
    method: 'GET',  
    headers: myHeaders,  
    redirect: 'follow'  
};
```

## Unirest

```
Unirest.setTimeouts(0, 0);  
HttpResponse<String> response = Unirest.get("https://v1.basketball.api-sports.io/1  
    .header("x-rapidapi-key", "XXXXXXXXXXXXXXXXXXXXXXXXXXXX")  
    .header("x-rapidapi-host", "v1.basketball.api-sports.io")  
    .asString();
```

# Javascript

## Fetch

```
var myHeaders = new Headers();  
myHeaders.append("x-rapidapi-key", "XXXXXXXXXXXXXXXXXXXXXXXXXXXX");  
myHeaders.append("x-rapidapi-host", "v1.basketball.api-sports.io");  
  
var requestOptions = {  
    method: 'GET',  
    headers: myHeaders,
```

```
    redirect: 'follow'
  };

  fetch("https://v1.basketball.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://v1.basketball.api-sports.io/leagues",
  "method": "GET",
  "timeout": 0,
  "headers": {
    "x-rapidapi-key": "XxXxXxXxXxXxXxXxXxXxXxXx",
    "x-rapidapi-host": "v1.basketball.api-sports.io"
  },
};

$.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://v1.basketball.api-sports.io/leagues");
xhr.setRequestHeader("x-rapidapi-key", "XxXxXxXxXxXxXxXxXxXxXxXx");
xhr.setRequestHeader("x-rapidapi-host", "v1.basketball.api-sports.io");

xhr.send();
```

# NodeJs

## Axios

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

var config = {
  method: 'get',
  url: 'https://v1.basketball.api-sports.io/leagues',
  headers: {
    'x-rapidapi-key': 'XXXXXXXXXXXXXXXXXXXXXXXXX',
    'x-rapidapi-host': 'v1.basketball.api-sports.io'
  }
};

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': 'v1.basketball.api-sports.io',
  'path': '/leagues',
  'headers': {
    'x-rapidapi-key': 'XXXXXXXXXXXXXXXXXXXXXXXXX',
    'x-rapidapi-host': 'v1.basketball.api-sports.io'
  },
  '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://v1.basketball.api-sports.io/leagues',  
  'headers': {  
    'x-rapidapi-key': 'XxXxXxXxXxXxXxXxXxXxXxXx',  
    'x-rapidapi-host': 'v1.basketball.api-sports.io'  
  }  
};  
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://v1.basketball.api-sports.io/leagues')  
  .headers({  
    'x-rapidapi-key': 'XxXxXxXxXxXxXxXxXxXxXxXx',  
    'x-rapidapi-host': 'v1.basketball.api-sports.io'  
  })  
  .end(function (res) {  
    if (res.error) throw new Error(res.error);  
    console.log(res.raw_body);  
  });
```

# Objective-c

## NSURLSession

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

dispatch_semaphore_t sema = dispatch_semaphore_create(0);

NSMutableURLRequest *request = [NSMutableURLRequest requestWithURL:[NSURL URLWithString:
    cachePolicy:NSURLRequestUseProtocolCachePolicy
    timeoutInterval:10.0];
NSDictionary *headers = @{
    @"x-rapidapi-key": @"XXXXXXXXXXXXXXXXXXXXXXXXXXXX",
    @"x-rapidapi-host": @"v1.basketball.api-sports.io"
};

[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:data
            error:&parseError];
        NSLog(@"%@", responseDictionary);
        dispatch_semaphore_signal(sema);
    }
}];

[dataTask resume];
dispatch_semaphore_wait(sema, DISPATCH_TIME_FOREVER);
```

## Cohttp

# Php

## 15/67

```

CURLOPT_HTTPHEADER => array(
    'x-rapidapi-key: XXXXXXXXXXXXXXXXXXXX',
    'x-rapidapi-host: v1.basketball.api-sports.io'
),
));

$response = curl_exec($curl);

curl_close($curl);
echo $response;

```

## Request2

```

<?php
require_once 'HTTP/Request2.php';
$request = new HTTP_Request2();
$request->setUrl('https://v1.basketball.api-sports.io/leagues');
$request->setMethod(HTTP_Request2::METHOD_GET);
$request->setConfig(array(
    'follow_redirects' => TRUE
));
$request->setHeader(array(
    'x-rapidapi-key' => 'XXXXXXXXXXXXXXXXXXXXXXXX',
    'x-rapidapi-host' => 'v1.basketball.api-sports.io'
));
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://v1.basketball.api-sports.io/leagues');
$request->setRequestMethod('GET');
$request->setHeaders(array(

```



```
'x-rapidapi-host' => 'v1.basketball.api-sports.io',  
'x-rapidapi-key' => 'XXXXXXXXXXXXXXXXXXXXXXXXX'  
));  
$client->enqueue($request)->send();  
$response = $client->getResponse();  
echo $response->getBody();
```

## PowerShell

RestMethod

```
$headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"  
$headers.Add("x-rapidapi-key", "XXXXXXXXXXXXXXXXXXXXXXXXX")  
$headers.Add("x-rapidapi-host", "v1.basketball.api-sports.io")  
  
$response = Invoke-RestMethod 'https://v1.basketball.api-sports.io/leagues' -Method  
$response | ConvertTo-Json
```

## Python

http.client

```
import http.client  
  
conn = http.client.HTTPSConnection("v1.basketball.api-sports.io")  
  
headers = {  
    'x-rapidapi-host': "v1.basketball.api-sports.io",  
    'x-rapidapi-key': "XXXXXXXXXXXXXXXXXXXXXXXXX"  
}  
  
conn.request("GET", "/leagues", headers=headers)  
  
res = conn.getresponse()  
data = res.read()
```

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

### Requests

```
url = "https://v1.basketball.api-sports.io/leagues"

payload={}
headers = {
    'x-rapidapi-key': 'XxXxXxXxXxXxXxXxXxXxXxXx',
    'x-rapidapi-host': 'v1.basketball.api-sports.io'
}

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

print(response.text)
```

## Ruby

### Net::HTTP

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

url = URI("https://v1.basketball.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-rapidapi-host"] = 'v1.basketball.api-sports.io'
request["x-rapidapi-key"] = 'XxXxXxXxXxXxXxXxXxXxXxXx'

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

## Shell

### Httpie

```
http --follow --timeout 3600 GET 'https://v1.basketball.api-sports.io/leagues' \
  x-rapidapi-key: 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX' \
  x-rapidapi-host: 'v1.basketball.api-sports.io'
```

### wget

```
wget --no-check-certificate --quiet \
  --method GET \
  --timeout=0 \
  --header 'x-rapidapi-key: XXXXXXXXXXXXXXXXXXXXXXXX' \
  --header 'x-rapidapi-host: v1.basketball.api-sports.io' \
  'https://v1.basketball.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://v1.basketball.api-sports.io/leagues")!)
request.addValue("XXXXXXXXXXXXXXXXXXXXXXXXXXXX", forHTTPHeaderField: "x-rapidapi-key")
request.addValue("v1.basketball.api-sports.io", forHTTPHeaderField: "x-rapidapi-host")

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

### (1.5.6)

- Endpoint `leagues`
  - Add field `coverage`
- Endpoint `games`
  - Add endpoint `games/statistics/teams`
  - Add endpoint `games/statistics/players`
  - Add field `venue`
- Add endpoint `players`

## 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

Our widgets are completely **free** and work with all our plans including the free plan.

To integrate the widgets to your site you just have to `copy/paste` the code provided and fill in the tags needed for the widget to work properly. If you integrate several widgets on the same page a single theme will be applied for all widgets. Also you will only have to integrate the **script** tag once.

## For all widgets the following tags are needed :

`data-host` : **v1.basketball.api-sports.io** or **api-basketball.p.rapidapi.com** depending on whether you have subscribed with us or RapidApi.

`data-key` : Indicate your API-KEY obtained on our Dashboard or on RapidApi.

`data-theme` : If you leave the field empty, the **default** theme will be applied, otherwise the possible values are **grey** or **dark**. It is also possible to indicate **false** which will not display any theme and lets you customize the widget with your own css.

`data-show-errors` : By default **false**, used for debugging, with a value of **true** it allows to display the errors.

Widgets use the requests associated with your account and therefore they will stop working if your daily limit is reached. You can track all requests made directly in the dashboard.

## Security :

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 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.

## 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

## Tutorials :

- [HOW CUSTOM API-FOOTBALL WIDGETS](#)
- [HOW TO CHANGE WIDGET ATTRIBUTES DYNAMICALLY](#)

Changelog :

2.0.3

- Starting version with widgets `Games` and `Standings`

Sources :

All the sources to make your own CSS can be downloaded here :

- [2.0.3 files](#)

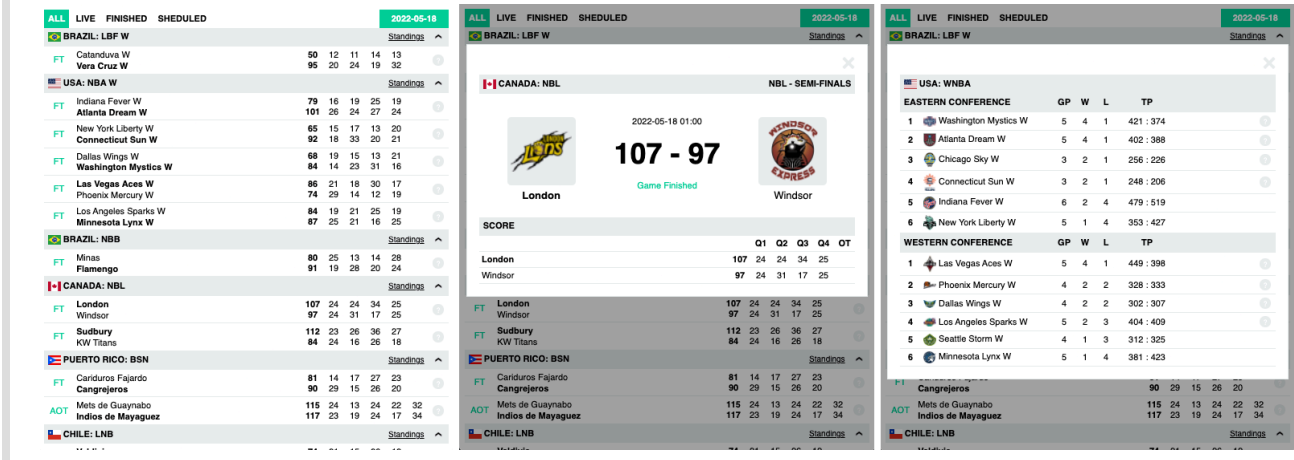
Games

Display the list of matches grouped by competition according to the parameters used.

The matches are automatically updated according to the selected frequency `data-refresh`.

You can find all the leagues ids on our [Dashboard](#).

Example of the widget with the default theme



QUERY PARAMETERS

<code>data-host</code> <code>required</code>	string Enum: <code>"v1.basketball.api-sports.io"</code> <code>"api-basketball.p.rapidapi.com"</code>
<code>data-key</code> <code>required</code>	string Your Api Key
<code>data-refresh</code>	integer <code>&gt;15</code>



Number in seconds corresponding to the desired data update frequency. If you indicate **0** or leave this field empty the data will not be updated automatically

---

data-date

stringYYYY-MM-DD

Fill in the desired date. If empty the current date is automatically applied

---

data-league

integer

Fill in the desired league id

---

data-season

integer = 4 characters YYYY

Fill in the desired season

---

data-theme

string

If you leave the field empty, the **default** theme will be applied, otherwise the possible values are **grey** or **dark**

---

data-show-toolbar

string

Enum: **true** **false**

Displays the toolbar allowing to change the view between the current, finished or upcoming fixtures and to change the date

---

data-show-logos

string

Enum: **true** **false**

If **true** display teams logos

---

data-modal-game

string

Enum: **true** **false**

If **true** allows to load a modal containing all the details of the fixture

---

data-modal-standings

string

Enum: **true** **false**

If **true** allows to load a modal containing the standings

---

data-modal-show-logos

string

Enum: **true** **false**

If **true** display teams logos and players images in the modal

---

data-show-errors

string

Enum: **true** **false**

By default **false**, used for debugging, with a value of **true** it allows to display the errors

---

GET /widgets/Games

## Request samples

**Html**

Copy

```
<div id="wg-api-basketball-games"
  data-host="v1.basketball.api-sports.io"
  data-key="Your-API-Key-Here"
  data-date=""
  data-league=""
  data-season=""
  data-theme=""
  data-refresh="15"
  data-show-toolbar="true"
  data-show-errors="false"
  data-show-logos="true"
  data-modal-game="true"
  data-modal-standings="true"
  data-modal-show-logos="true">
</div>
<script
  type="module"
  src="https://widgets.api-sports.io/2.0.3/widgets.js">
</script>
```

## Standings

Display the ranking of a competition according to the parameters used.

You can find all the leagues ids on our [Dashboard](#).

Example of the widget with the available themes

USA: NBA - REGULAR SEASON									
WESTERN CONFERENCE					GP	W	L	TP	
1		Phoenix Suns	82	64	18	9416	: 8801		?
2		Memphis Grizzlies	82	56	26	9480	: 9014		?
3		Golden State Warriors	82	53	29	9102	: 8648		?
4		Dallas Mavericks	82	52	30	8858	: 8587		?
5		Utah Jazz	82	49	33	9316	: 8821		?
6		Denver Nuggets	82	48	34	9243	: 9054		?
7		Minnesota Timberwolves	82	46	36	9507	: 9291		?
8		New Orleans Pelicans	82	36	46	8962	: 9043		?
9		Los Angeles Clippers	82	42	40	8890	: 8888		?
10		San Antonio Spurs	82	34	48	9279	: 9269		?
11		Los Angeles Lakers	82	33	49	9192	: 9442		?
12		Sacramento Kings	82	30	52	9045	: 9493		?
13		Portland Trail Blazers	82	27	55	8711	: 9439		?
14		Oklahoma City Thunder	82	24	58	8506	: 9170		?
15		Houston Rockets	82	20	62	8997	: 9692		?
EASTERN CONFERENCE					GP	W	L	TP	
1		Miami Heat	82	53	29	9022	: 8657		?
2		Boston Celtics	82	51	31	9164	: 8567		?
3		Milwaukee Bucks	82	51	31	9470	: 9195		?
4		Philadelphia 76ers	82	51	31	9015	: 8801		?
5		Toronto Raptors	82	48	34	8970	: 8782		?
6		Chicago Bulls	82	46	36	9152	: 9184		?

USA: NBA - REGULAR SEASON									
WESTERN CONFERENCE					GP	W	L	TP	
1		Phoenix Suns	82	64	18	9416	: 8801		?
2		Memphis Grizzlies	82	56	26	9480	: 9014		?
3		Golden State Warriors	82	53	29	9102	: 8648		?
4		Dallas Mavericks	82	52	30	8858	: 8587		?
5		Utah Jazz	82	49	33	9316	: 8821		?
6		Denver Nuggets	82	48	34	9243	: 9054		?
7		Minnesota Timberwolves	82	46	36	9507	: 9291		?
8		New Orleans Pelicans	82	36	46	8962	: 9043		?
9		Los Angeles Clippers	82	42	40	8890	: 8888		?
10		San Antonio Spurs	82	34	48	9279	: 9269		?
11		Los Angeles Lakers	82	33	49	9192	: 9442		?
12		Sacramento Kings	82	30	52	9045	: 9493		?
13		Portland Trail Blazers	82	27	55	8711	: 9439		?
14		Oklahoma City Thunder	82	24	58	8506	: 9170		?
15		Houston Rockets	82	20	62	8997	: 9692		?
EASTERN CONFERENCE					GP	W	L	TP	
1		Miami Heat	82	53	29	9022	: 8657		?
2		Boston Celtics	82	51	31	9164	: 8567		?
3		Milwaukee Bucks	82	51	31	9470	: 9195		?
4		Philadelphia 76ers	82	51	31	9015	: 8801		?
5		Toronto Raptors	82	48	34	8970	: 8782		?
6		Chicago Bulls	82	46	36	9152	: 9184		?

USA: NBA - REGULAR SEASON									
WESTERN CONFERENCE					GP	W	L	TP	
1		Phoenix Suns	82	64	18	9416	: 8801		?
2		Memphis Grizzlies	82	56	26	9480	: 9014		?
3		Golden State Warriors	82	53	29	9102	: 8648		?
4		Dallas Mavericks	82	52	30	8858	: 8587		?
5		Utah Jazz	82	49	33	9316	: 8821		?
6		Denver Nuggets	82	48	34	9243	: 9054		?
7		Minnesota Timberwolves	82	46	36	9507	: 9291		?
8		New Orleans Pelicans	82	36	46	8962	: 9043		?
9		Los Angeles Clippers	82	42	40	8890	: 8888		?
10		San Antonio Spurs	82	34	48	9279	: 9269		?
11		Los Angeles Lakers	82	33	49	9192	: 9442		?
12		Sacramento Kings	82	30	52	9045	: 9493		?
13		Portland Trail Blazers	82	27	55	8711	: 9439		?
14		Oklahoma City Thunder	82	24	58	8506	: 9170		?
15		Houston Rockets	82	20	62	8997	: 9692		?
EASTERN CONFERENCE					GP	W	L	TP	
1		Miami Heat	82	53	29	9022	: 8657		?
2		Boston Celtics	82	51	31	9164	: 8567		?
3		Milwaukee Bucks	82	51	31	9470	: 9195		?
4		Philadelphia 76ers	82	51	31	9015	: 8801		?
5		Toronto Raptors	82	48	34	8970	: 8782		?
6		Chicago Bulls	82	46	36	9152	: 9184		?

QUERY PARAMETERS

data-host  
required

string  
Enum: "v1.basketball.api-sports.io" "api-basketball.p.rapidapi.com"

data-key  
required

string  
Your Api Key

data-league

integer  
Fill in the desired league id

data-season  
required

integer = 4 characters YYYY  
Fill in the desired season

data-theme

string  
If you leave the field empty, the default theme will be applied, otherwise the possible values are grey or dark

data-show-errors

string  
Enum: true false  
By default false, used for debugging, with a value of true it allows to display the errors

data-show-logos

string  
Enum: true false  
If true display teams logos

GET /widgets/standings

## Request samples

Html

Copy

```
<div id="wg-api-basketball-standings"
  data-host="v1.basketball.api-sports.io"
  data-key="Your-API-Key-Here"
  data-league="12"
  data-season="2021-2022"
  data-theme=""
  data-show-errors="false"
  data-show-logos="true"
  class="wg_loader">
</div>
<script
  type="module"
  src="https://widgets.api-sports.io/2.0.3/widgets.js">
</script>
```

# Timezone

## timezone

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

This endpoint does not require any parameters.

### HEADER PARAMETERS

x-rapidapi-key	string
----------------	--------

required

Your RapidAPI Key

---

## Responses

&gt; 200 OK

GET /timezone

## Request samples

Php

Python

Node

JavaScript

Curl

Ruby

Copy

```
$client = new http\Client;
$request = new http\Client\Request;

$request->setRequestUrl('https://v1.basketball.api-sports.io/timezone');
$request->setRequestMethod('GET');
$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.basketball.api-sports.io',
    'x-rapidapi-key' => 'XXXXXXXXXXXXXXXXXXXXXXXXX'
));

$client->enqueue($request)->send();
$response = $client->getResponse();

echo $response->getBody();
```

## Response samples

200

Content type

application/json

Copy

Expand all

Collapse all

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

---

## Seasons

### seasons

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

This endpoint does not require any parameters.

#### HEADER PARAMETERS

---

x-rapidapi-key	string
<b>required</b>	Your RapidAPI Key

---

### Responses

> 200 OK

GET /seasons

## Request samples

Php

Python

Node

JavaScript

Curl

Ruby

Copy

```
$client = new http\Client;
$request = new http\Client\Request;

$request->setRequestUrl('https://v1.basketball.api-sports.io/seasons');
$request->setRequestMethod('GET');
$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.basketball.api-sports.io',
    'x-rapidapi-key' => 'XXXXXXXXXXXXXXXXXXXXXXXXX'
));

$client->enqueue($request)->send();
$response = $client->getResponse();

echo $response->getBody();
```

## Response samples

200

### Content type

application/json

Copy

Expand all

Collapse all

```
{
  "get": "seasons",
  "parameters": [ ],
  "errors": [ ],
  "results": 8,
```

```
- "response": [  
  "2015-2016",  
  "2016-2017",  
  2017,  
  "2017-2018",  
  2018,  
  "2018-2019",  
  2019,  
  "2019-2020"  
]
```

---

## Countries

### countries

Get the list of available countries.

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

All the parameters of this endpoint can be used together.

#### QUERY PARAMETERS

---

id	integer The id of the country
name	string Example: <code>name=USA</code> The name of the country
code	string <code>= 2 characters</code> Example: <code>code=EN</code> The code of the country



search

string >= 3 charactersExample: 

---

## HEADER PARAMETERS

---

x-rapidapi-key  
**required**

string  
Your RapidAPI Key

---

## Responses

> 200 OK

GET /countries

## Request samples

Php

Python

Node

JavaScript

Curl

Ruby

Use Cases

Copy

```
$client = new http\Client;
$request = new http\Client\Request;

$request->setRequestUrl('https://v1.basketball.api-sports.io/countries');
$request->setRequestMethod('GET');
$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.basketball.api-sports.io',
    'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXxXx'
));

$client->enqueue($request)->send();
$response = $client->getResponse();

echo $response->getBody();
```

## Response samples

200

**Content type**

application/json

Copy Expand all Collapse all

```
{
  "get": "countries",
  - "parameters": {
    "search": "usa"
  },
  "errors": [ ],
  "results": 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`

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

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.

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

You can find all the leagues ids on our [Dashboard](#).

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

## QUERY PARAMETERS

id	integer The id of the league
name	string Example: <code>name=NBA</code> The name of the league
country_id	integer The id of the country
country	string Example: <code>country=USA</code> The name of the country
type	string Enum: <code>"league"</code> <code>"cup"</code> Example: <code>type=league</code> The type of the league
season	string <code>[ 4 .. 9 ] characters</code> <code>YYYY</code> or <code>YYYY-YYYY</code> Example: <code>season=2021-2022</code> The season of the league
search	string <code>&gt;= 3 characters</code> Example: <code>search=NBA</code> The name of the league
code	string <code>= 2 characters</code> Example: <code>code=FR</code> The code of the country

## HEADER PARAMETERS

x-rapidapi-key <b>required</b>	string You rapidAPI Key
-----------------------------------	----------------------------

## Responses

> 200 OK

GET /leagues

Request samples

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

Copy

```
$client = new http\Client;
$request = new http\Client\Request;

$request->setRequestUrl('https://v1.basketball.api-sports.io/leagues');
$request->setRequestMethod('GET');
$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.basketball.api-sports.io',
    'x-rapidapi-key' => 'XXXXXXXXXXXXXXXXXXXXXXXXX'
));

$client->enqueue($request)->send();
$response = $client->getResponse();

echo $response->getBody();
```

Response samples

200

Content type  
application/json

Copy Expand all Collapse all

```
{
  "get": "leagues",
  - "parameters": {
    "id": "12",
    "season": "2023-2024"
  },
  "errors": [ ],
}
```

```
    "results": 1,
  -  "response": [
      +  { ... }
    ]
  }
```

# Teams

## teams

Get data about teams.

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

You can find all the teams ids on our [Dashboard](#).

**This endpoint requires at least one parameter.**

### QUERY PARAMETERS

id	integer The id of the team
name	string Example: <code>name=Denver Nuggets</code> The name of the team
country_id	integer The id of the country
country	string The name of the country
league	integer The id of the league

season

string [ 4 .. 9 ] characters YYYY or YYYY-YYYY

Example: `season=2021-2022`

The season of the league

search

string `>= 3 characters`Example: `search=Denver`

The name of the team

## HEADER PARAMETERS

`x-rapidapi-key`  
`required`

string

Your RapidAPI Key

## Responses

&gt; 200 OK

GET /teams

## Request samples

Php

Python

Node

JavaScript

Curl

Ruby

Use Cases

Copy

```
$client = new http\Client;
$request = new http\Client\Request;

$request->setRequestUrl('https://v1.basketball.api-sports.io/teams');
$request->setRequestMethod('GET');
$request->setQuery(new http\QueryString(array(
    'id' => '139'
)));

$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.basketball.api-sports.io',
    'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXxXx'
));
```

```
$client->enqueue($request)->send();
$response = $client->getResponse();

echo $response->getBody();
```

Response samples

200

Content type  
application/json

Copy Expand all Collapse all

```
{
  "get": "teams",
  - "parameters": {
    "name": "Denver Nuggets"
  },
  "errors": [ ],
  "results": 1,
  - "response": [
    + { ... }
  ]
}
```

statistics

QUERY PARAMETERS

league required	integer The id of the league
season required	string [ 4 .. 9 ] characters YYYY or YYYY-YYYY Example: season=2021-2022 The season of the league
team required	integer The id of the team

date

stringYYYY-MM-DD

Example: 

A Limit Date

---

HEADER PARAMETERS

---

x-rapidapi-key  
requiredstring  
Your RapidAPI Key

---

## Responses

&gt; 200 OK

GET /statistics

## Request samples

Php

Python

Node

JavaScript

Curl

Ruby

Use Cases

Copy

```
$client = new http\Client;
$request = new http\Client\Request;

$request->setRequestUrl('https://v1.basketball.api-sports.io/statistics');
$request->setRequestMethod('GET');
$request->setQuery(new http\QueryString(array(
    'season' => '2019-2020',
    'team' => '139',
    'league' => '12'
)));

$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.basketball.api-sports.io',
    'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXxXx'
));

$client->enqueue($request)->send();
```



```
$response = $client->getResponse();
```

```
echo $response->getBody();
```

## Response samples

**200**

### Content type

application/json

[Copy](#)[Expand all](#)[Collapse all](#)

```
{
  "get": "statistics",
  - "parameters": {
    "league": "12",
    "season": "2019-2020",
    "team": "139"
  },
  "errors": [ ],
  "results": 5,
  - "response": {
    + "league": { ... },
    + "country": { ... },
    + "team": { ... },
    + "games": { ... },
    + "points": { ... }
  }
}
```

---

## Players

### players

Get data about players.

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

**This endpoint requires at least one parameter.**

QUERY PARAMETERS

id	integer The id of the player
team	integer The id of the team
season	string <code>[ 4 .. 9 ] characters</code> <code>YYYY</code> or <code>YYYY-YYYY</code> Example: <code>season=2023-2024</code> A valid season
search	string <code>&gt;= 3 characters</code> Example: <code>search=Malith</code> The name of the player

HEADER PARAMETERS

x-rapidapi-key required	string Your RapidAPI Key
----------------------------	-----------------------------

Responses

> 200 OK

GET /players

Request samples

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

Copy

```
$client = new http\Client;
$request = new http\Client\Request;

$request->setRequestUrl('https://v1.basketball.api-sports.io/players');
$request->setRequestMethod('GET');
$request->setQuery(new http\QueryString(array(
    'team' => '1'
)));

$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.basketball.api-sports.io',
    'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXxXx'
));

$client->enqueue($request)->send();
$response = $client->getResponse();

echo $response->getBody();
```

## Response samples

200

### Content type

application/json

Copy

Expand all

Collapse all

```
{
  "get": "players",
  - "parameters": {
    "team": "1",
    "season": "2023-2024"
  },
  "errors": [ ],
  "results": 16,
```

}

## standings

Return a table of one or more rankings according to the league / cup. Some competitions have several rankings in a year, regular season, pre season etc...

Standings are updated every hours

<https://api-sports.io/documentation/basketball/v1>

league  
required

integer  
The id of the league

---

season  
required

string [ 4 .. 9 ] characters YYYY or YYYY-YYYY  
Example: `season=2021-2022`  
The season of the league

---

team

integer  
The id of the team

---

stage

string  
Example: `stage=NBA - Regular Season`  
A valid stage

---

group

string  
Example: `group=Eastern Conference`  
A valid group

---

#### HEADER PARAMETERS

---

x-rapidapi-key  
required

string  
Your RapidAPI Key

---

## Responses

> 200 OK

GET /standings

## Request samples

Php

Python

Node

JavaScript

Curl

Ruby

Use Cases

Copy

```
$client = new http\Client;  
$request = new http\Client\Request;
```

```
$request->setRequestUrl('https://v1.basketball.api-sports.io/standings');
$request->setRequestMethod('GET');
$request->setQuery(new http\QueryString(array(
    'league' => '12',
    'season' => '2019-2020'
)));

$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.basketball.api-sports.io',
    'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXxXx'
));

$client->enqueue($request)->send();
$response = $client->getResponse();

echo $response->getBody();
```

## Response samples

**200**

### Content type

application/json

Copy

Expand all

Collapse all

```
{
  "get": "standings",
  - "parameters": {
    "league": "12",
    "season": "2019-2020",
    "team": "137"
  },
  "errors": [ ],
  "results": 1,
  - "response": [
    + [ ... ]
  ]
}
```

Get the list of available stages for a league to be used in the standings endpoint.

## QUERY PARAMETERS

league required	integer The id of the league
season required	string [ 4 .. 9 ] characters YYYY or YYYY-YYYY Example: season=2021-2022 The season of the league

## HEADER PARAMETERS

x-rapidapi-key required	string Your RapidAPI Key
----------------------------	-----------------------------

## Responses

> 200 OK

GET /standings/stages

## Request samples

Php

Python

Node

JavaScript

Curl

Ruby

Copy

```
$client = new http\Client;
$request = new http\Client\Request;

$request->setRequestUrl('https://v1.basketball.api-sports.io/standings/stages');
$request->setRequestMethod('GET');
$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.basketball.api-sports.io',
    'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXxXx'
));
```

```
$client->enqueue($request)->send();
$response = $client->getResponse();

echo $response->getBody();
```

## Response samples

200

### Content type

application/json

Copy Expand all Collapse all

```
{
  "get": "standings/stages",
  - "parameters": {
    "league": "12",
    "season": "2019-2020"
  },
  "errors": [ ],
  "results": 1,
  - "response": [
    "NBA - Regular Season"
  ]
}
```

## standings/groups

Get the list of available groups for a league to be used in the standings endpoint.

### QUERY PARAMETERS

league  
required

integer  
The id of the league

season  
required

string [ 4 .. 9 ] characters YYYY or YYYY-YYYY  
Example: `season=2021-2022`



The season of the league

HEADER PARAMETERS

x-rapidapi-key	string
required	Your RapidAPI Key

Responses

> 200 OK

— 201 Created

GET /standings/groups

Request samples

- Php
- Python
- Node
- JavaScript
- Curl
- Ruby

Copy

```
$client = new http\Client;
$request = new http\Client\Request;

$request->setRequestUrl('https://v1.basketball.api-sports.io/standings/groups');
$request->setRequestMethod('GET');
$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.basketball.api-sports.io',
    'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXxXx'
));

$client->enqueue($request)->send();
$response = $client->getResponse();

echo $response->getBody();
```

Response samples

200

**Content type**

application/json

Copy Expand all Collapse all

```
{
  "get": "standings/groups",
  - "parameters": {
    "league": "12",
    "season": "2019-2020"
  },
  "errors": [ ],
  "results": 8,
  - "response": [
    "Western Conference",
    "Eastern Conference",
    "Atlantic",
    "Southeast",
    "Central",
    "Northwest",
    "Pacific",
    "Southwest"
  ]
}
```

## Games

### games

For all requests to games you can add the query parameter `timezone` to your request in order to retrieve the list of games 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 status

- NS : Not Started
- Q1 : Quarter 1 (In Play)
- Q2 : Quarter 2 (In Play)
- Q3 : Quarter 3 (In Play)
- Q4 : Quarter 4 (In Play)
- OT : Over Time (In Play)
- BT : Break Time (In Play)
- HT : Halftime (In Play)
- FT : Game Finished (Game Finished)
- AOT : After Over Time (Game Finished)
- POST : Game Postponed
- CANC : Game Cancelled
- SUSP : Game Suspended
- AWD : Game Awarded
- ABD : Game Abandoned

Games are updated every 15 seconds

**This endpoint requires at least one parameter.**

QUERY PARAMETERS

id	integer The id of the game
date	stringYYYY-MM-DD Example: <code>date=2019-11-23</code> A valid date
league	integer The id of the league
season	string [ 4 .. 9 ] characters YYYY or YYYY-YYYY Example: <code>season=2021-2022</code> The season of the league
team	integer The id of the team
timezone	string Example: <code>timezone=Europe/London</code> A valid timezone

HEADER PARAMETERS

x-rapidapi-key  
required

string  
Your RapidAPI Key

---

## Responses

> 200 OK

GET /games

## Request samples

Php

Python

Node

JavaScript

Curl

Ruby

Use Cases

Copy

```
$client = new http\Client;
$request = new http\Client\Request;

$request->setRequestUrl('https://v1.basketball.api-sports.io/games');
$request->setRequestMethod('GET');
$request->setQuery(new http\QueryString(array(
    'date' => '2019-11-23'
)));

$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.basketball.api-sports.io',
    'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXxXx'
));

$client->enqueue($request)->send();
$response = $client->getResponse();

echo $response->getBody();
```

## Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
  "get": "games",
  - "parameters": {
    "league": "12",
    "date": "2019-11-23",
    "team": "134",
    "timezone": "europe/london",
    "season": "2019-2020"
  },
  "errors": [ ],
  "results": 1,
  - "response": [
    + { ... }
  ]
}
```

# teams statistics

Get teams statistics from one or several games ids.

- Statistics are updated every 30-120 seconds
- This endpoint need at least one parameter.**

QUERY PARAMETERS

id	string The id of the game
ids	stringMaximum of 20 games ids Value: "id-id-id" One or more games ids

HEADER PARAMETERS

x-rapidapi-key	string
----------------	--------

required

Your RapidAPI Key

---

## Responses

&gt; 200 OK

GET /games/statistics/teams

## Request samples

Php

Python

Node

JavaScript

Curl

Ruby

Use Cases

Copy

```
$client = new http\Client;
$request = new http\Client\Request;

$request->setRequestUrl('https://v1.basketball.api-sports.io/games/statistics/te
$request->setRequestMethod('GET');
$request->setQuery(new http\QueryString(array(
    'id' => '391053'
)));

$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.basketball.api-sports.io',
    'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXxXx'
));

$client->enqueue($request)->send();
$response = $client->getResponse();

echo $response->getBody();
```

## Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
{
  "get": "games",
  - "parameters": {
    "id": "391053"
  },
  "errors": [ ],
  "results": 2,
  - "response": [
    + { ... },
    + { ... }
  ]
}
```

# players statistics

Get players statistics from one or several games ids.

Also possible to get all statistics from a player `id` and a `season`.

- Statistics are updated every 30-120 seconds
- This endpoint need at least one parameter.**

## QUERY PARAMETERS

id	string The id of the game
ids	stringMaximum of 20 games ids Value: "id-id-id" One or more games ids
player	integer The id of the player
season	string

## A valid season

## HEADER PARAMETERS

x-rapidapi-key required	string Your RapidAPI Key
----------------------------	-----------------------------

## Responses

&gt; 200 OK

GET /games/statistics/players

## Request samples

Php

Python

Node

JavaScript

Curl

Ruby

Use Cases

Copy

```
$client = new http\Client;
$request = new http\Client\Request;

$request->setRequestUrl('https://v1.basketball.api-sports.io/games/statistics/pl
$request->setRequestMethod('GET');
$request->setQuery(new http\QueryString(array(
    'id' => '391053'
)));

$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.basketball.api-sports.io',
    'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXxXx'
));

$client->enqueue($request)->send();
$response = $client->getResponse();

echo $response->getBody();
```



Response samples

200

Content type

application/json

Copy   Expand all   Collapse all

```
{
  "get": "games",
  - "parameters": {
    "id": "391053"
  },
  "errors": [ ],
  "results": 18,
  - "response": [
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... }
  ]
}
```

h2h

Get heads to heads between two teams.

QUERY PARAMETERS

<div>h2h</div> <div>required</div>	<div>stringid-id</div> <div>Example: <div>h2h=132-134</div></div> <div>The ids of the teams</div>
<div>date</div>	<div>stringYYYY-MM-DD</div> <div>Example: <div>date=2019-12-05</div></div> <div>A valid date</div>
<div>league</div>	<div>integer</div> <div>The id of the league</div>
<div>season</div>	<div>string<div>[ 4 .. 9 ] characters</div>YYYY or YYYY-YYYY</div> <div>Example: <div>season=2021-2022</div></div> <div>The season of the league</div>
<div>timezone</div>	<div>string</div> <div>Example: <div>timezone=Europe/London</div></div> <div>A valid timezone</div>

HEADER PARAMETERS

<div>x-rapidapi-key</div> <div>required</div>	<div>string</div> <div>Your RapidAPI Key</div>
-----------------------------------------------	------------------------------------------------

Responses

> 200 OK

GET /games/h2h

Request samples

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

Copy

```
$client = new http\Client;
$request = new http\Client\Request;

$request->setRequestUrl('https://v1.basketball.api-sports.io/games');
$request->setRequestMethod('GET');
$request->setQuery(new http\QueryString(array(
    'h2h' => '132-134'
)));

$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.basketball.api-sports.io',
    'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXxXx'
));

$client->enqueue($request)->send();
$response = $client->getResponse();

echo $response->getBody();
```

## Response samples

**200**

### Content type

application/json

Copy

Expand all

Collapse all

```
{
  "get": "games",
  - "parameters": {
    "league": "12",
    "h2h": "132-134",
    "season": "2019-2020"
  },
  "errors": [ ],
  "results": 4,
  - "response": [
    + { ... }
  ]
}
```

---

## Odds

### odds

Get odds from games or leagues.

We provide pre-match odds between 1 and 7 days before the game.

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

■ Odds are updated once a day

#### QUERY PARAMETERS

---

league

integer

The id of the league

season

string [ 4 .. 9 ] characters YYYY or YYYY-YYYY

Example: `season=2021-2022`

The season of the league

game	integer The id of the game
bookmaker	integer The id of the bookmaker
bet	integer The id of the bet

#### HEADER PARAMETERS

x-rapidapi-key required	string Your RapidAPI Key
----------------------------	-----------------------------

## Responses

> 200 OK

GET /odds

## Request samples

[Php](#)[Python](#)[Node](#)[JavaScript](#)[Curl](#)[Ruby](#)[Use Cases](#)[Copy](#)

```
$client = new http\Client;
$request = new http\Client\Request;

$request->setRequestUrl('https://v1.basketball.api-sports.io/odds');
$request->setRequestMethod('GET');
$request->setQuery(new http\QueryString(array(
    'season' => '2019-2020',
    'bet' => '1',
    'bookmaker' => '6',
    'game' => '1912',
    'league' => '12'
)));
```

```
$request->setHeaders(array(  
    'x-rapidapi-host' => 'v1.basketball.api-sports.io',  
    'x-rapidapi-key' => 'XXXXXXXXXXXXXXXXXXXXXXXXX'  
));  
  
$client->enqueue($request)->send();  
$response = $client->getResponse();  
  
echo $response->getBody();
```

## Response samples

**200**

### Content type

application/json

Copy

Expand all

Collapse all

```
{  
  "get": "odds",  
  - "parameters": {  
    "bet": "2",  
    "game": "1912"  
  },  
  "errors": [ ],  
  "results": 1,  
  - "response": [  
    + { ... }  
  ]  
}
```

## bookmakers

Get all available bookmakers.

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

### QUERY PARAMETERS

id	integer The id of the bookmaker
search	string <span>&gt;= 3 characters</span> Example: <span>search=Bwin</span> The name of the bookmaker

## HEADER PARAMETERS

x-rapidapi-key <span>required</span>	string Your RapidAPI Key
-----------------------------------------	-----------------------------

## Responses

> 200 OK

GET /bookmakers

## Request samples

[Php](#)[Python](#)[Node](#)[JavaScript](#)[Curl](#)[Ruby](#)[Use Cases](#)[Copy](#)

```
$client = new http\Client;
$request = new http\Client\Request;

$request->setRequestUrl('https://v1.basketball.api-sports.io/bookmakers');
$request->setRequestMethod('GET');
$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.basketball.api-sports.io',
    'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXxXx'
));

$client->enqueue($request)->send();
$response = $client->getResponse();
```

```
echo $response->getBody();
```

Response samples

200

Content type  
application/json

Copy   Expand all   Collapse all

```
{
  "get": "bookmakers",
  "parameters": [ ],
  "errors": [ ],
  "results": 15,
  - "response": [
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... },
    + { ... }
  ]
}
```

bets

Get all available bets.



All bets `id` can be used in endpoint odds as filters

## QUERY PARAMETERS

---

id	integer The id of the bet
----	------------------------------

---

search	string <code>&gt;= 3 characters</code> Example: <code>search=under</code> The name of the bet
--------	-----------------------------------------------------------------------------------------------------

---

## HEADER PARAMETERS

---

x-rapidapi-key <i>required</i>	string Your RapidAPI Key
-----------------------------------	-----------------------------

---

## Responses

> 200 OK

GET /bets

## Request samples

[Php](#)[Python](#)[Node](#)[JavaScript](#)[Curl](#)[Ruby](#)[Use Cases](#)[Copy](#)

```
$client = new http\Client;
$request = new http\Client\Request;

$request->setRequestUrl('https://v1.basketball.api-sports.io/bets');
$request->setRequestMethod('GET');
$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.basketball.api-sports.io',
    'x-rapidapi-key' => 'XXXXXXXXXXXXXXXXXXXXXXXXX'
));
```

```
$client->enqueue($request)->send();  
$response = $client->getResponse();  
  
echo $response->getBody();
```

Response samples

200

Content type

application/json

Copy    Expand all    Collapse all

[illegible]