API-FORMULA-1 (1.3.1)

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

Introduction

Welcome to Api-Formula-1! You can use our API to access all API endpoints, which can get information about Formula-1 competition.

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 uses 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-formula-1.p.rapidapi.com/

API-SPORTS : https://v1.formula-1.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 `XxXxXxXxXxXxXxXxXxXxXxXx` 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.formula-1.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
        }
```

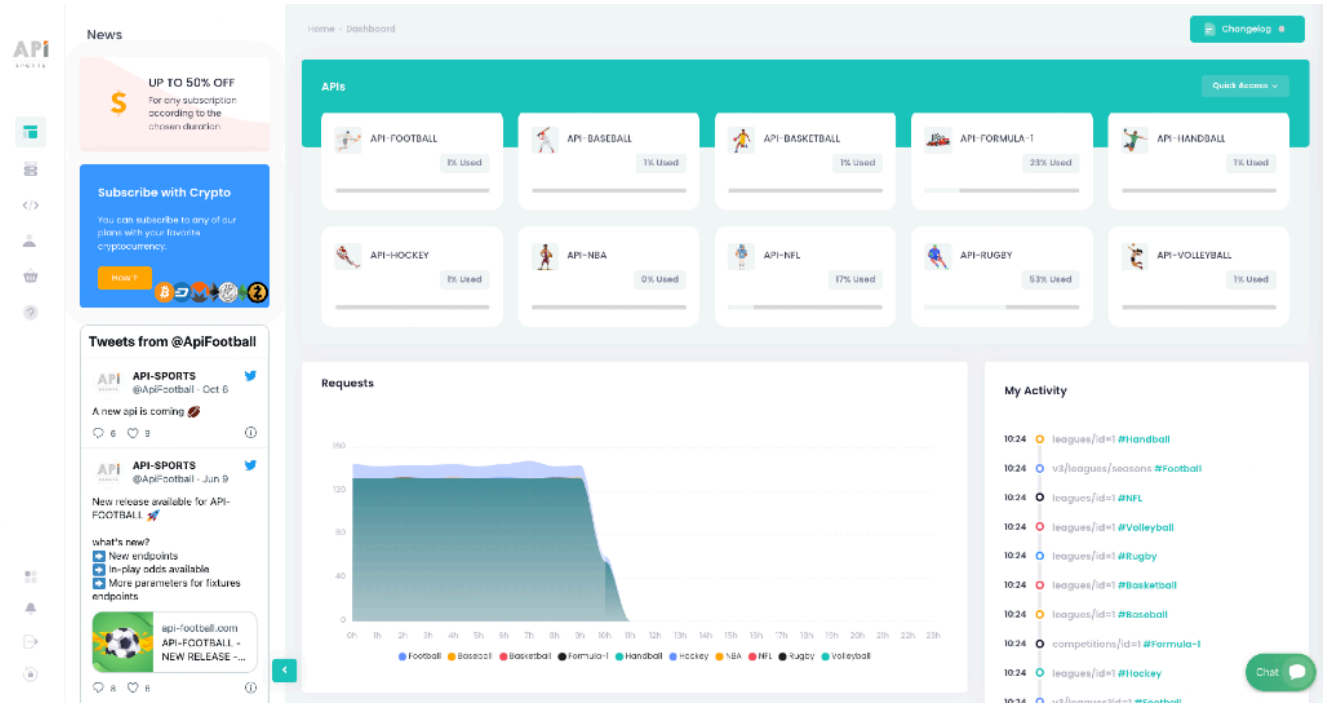
```
    }
}
```

Headers sent as response

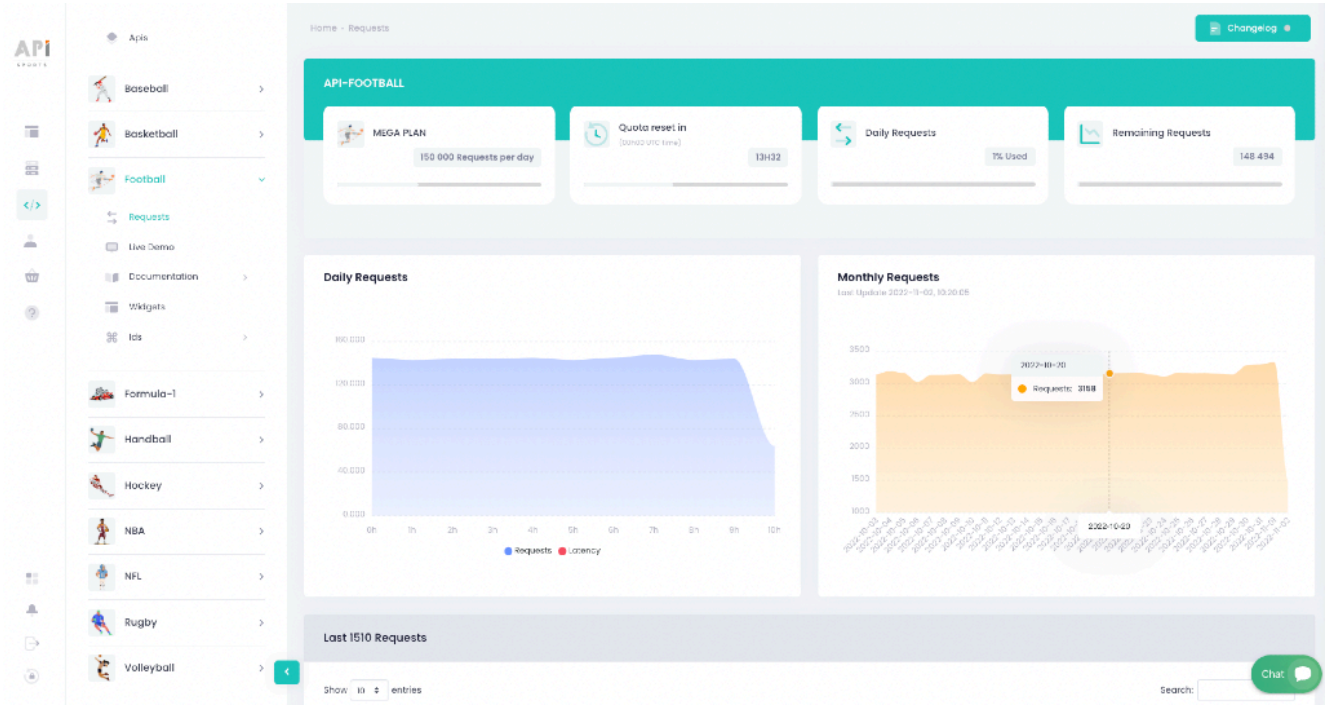
When consuming our API, you will always receive the following headers appended to the response:

- `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.

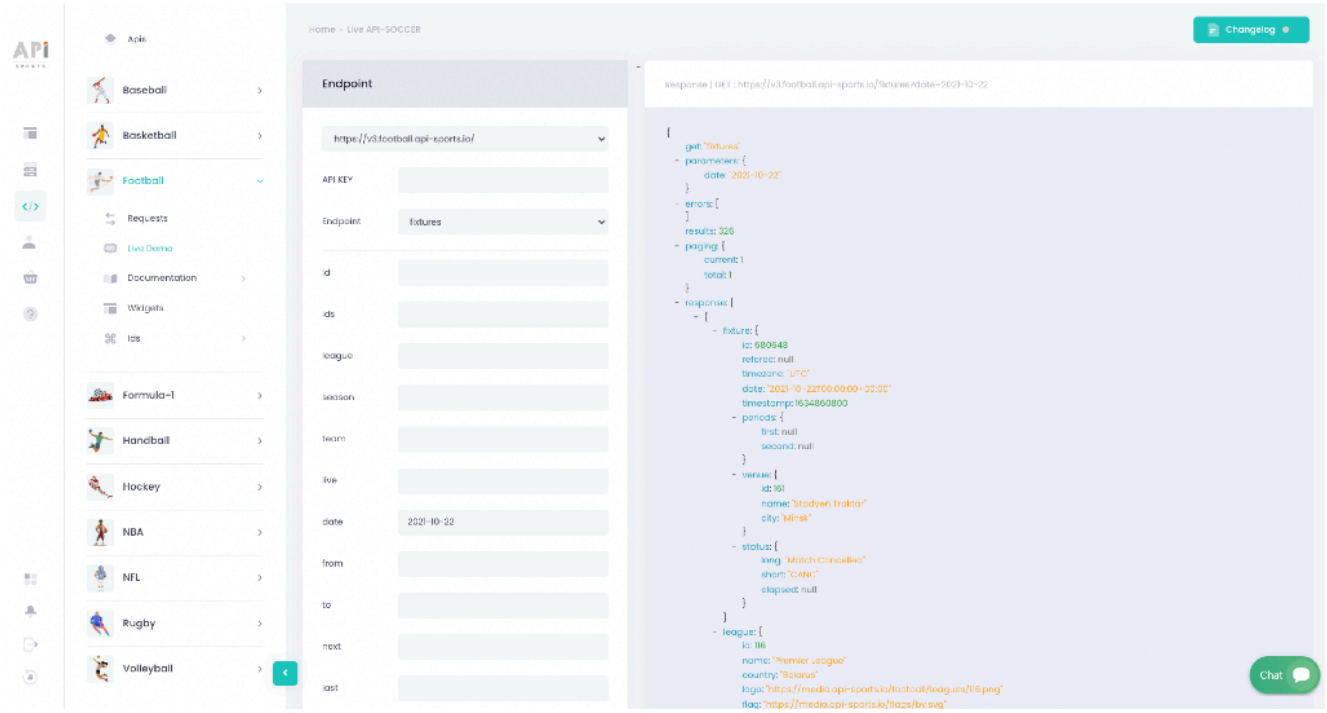
Dashboard



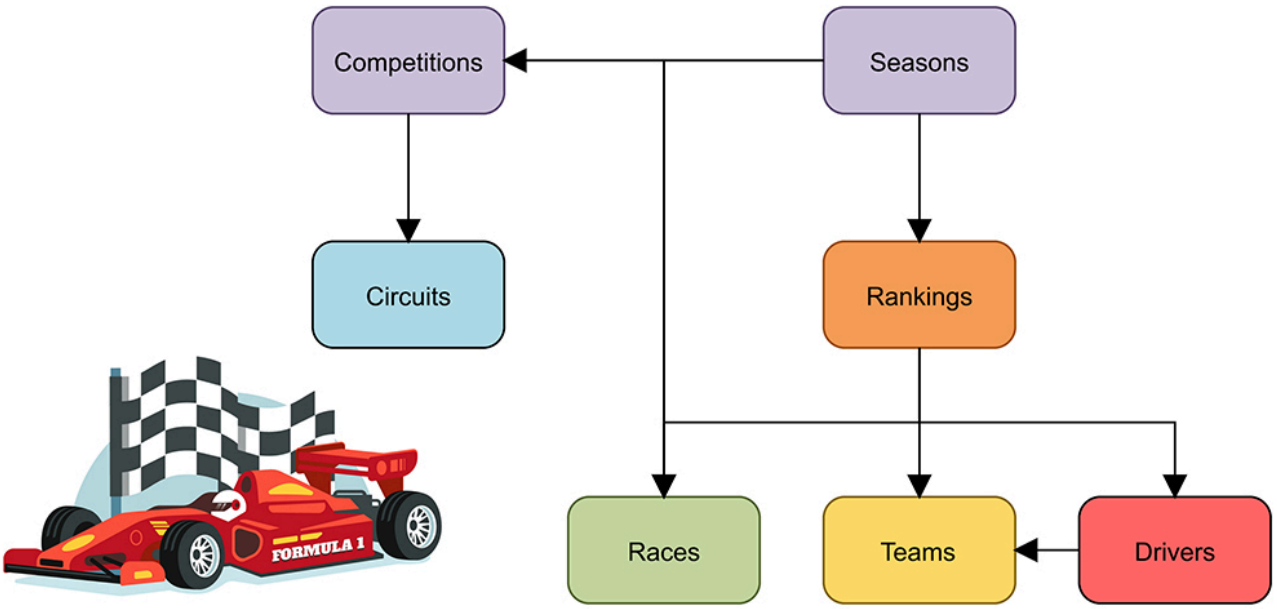
Requests



Live tester



Architecture



Logos / Images

Calls to logos/images do not count towards your daily quota. 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 `competitions` or `races` for example. In all the sample scripts we will use the `competitions` endpoint as example.

Also you will have to replace `XxXxXxXxXxXxXxXxXxXxXx` 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.formula-1.api-sports.io/competit
  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: XxXxXxXxXxXxXxXxXxXxXx")
  headers = curl_slist_append(headers, "x-rapidapi-host: v1.formula-1.api-sports.i
  curl_easy_setopt(curl, CURLOPT_HTTPHEADER, headers);
  res = curl_easy_perform(curl);
}
curl_easy_cleanup(curl);
```

C#

`RestSharp`

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

```
request.AddHeader("x-rapidapi-host", "v1.formula-1.api-sports.io");
IRestResponse response = client.Execute(request);
Console.WriteLine(response.Content);
```

cURL

`Curl`

```
curl --request GET \
     --url https://v1.formula-1.api-sports.io/competitions \
     --header 'x-rapidapi-host: v1.formula-1.api-sports.io' \
     --header 'x-rapidapi-key: XxXxXxXxXxXxXxXxXxXxXxXx'
```

Dart

`http`

```
var headers = {
  'x-rapidapi-key': 'XxXxXxXxXxXxXxXxXxXxXxXx',
  'x-rapidapi-host': 'v1.formula-1.api-sports.io'
};
var request = http.Request('GET', Uri.parse('https://v1.formula-1.api-sports.io/co

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.formula-1.api-sports.io/competitions"
  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", "XxXxXxXxXxXxXxXxXxXxXxXx")
  req.Header.Add("x-rapidapi-host", "v1.formula-1.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", "XxXxXxXxXxXxXxXxXxXxXxXx");
myHeaders.append("x-rapidapi-host", "v1.formula-1.api-sports.io");

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

Unirest

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

Javascript

Fetch

```
var myHeaders = new Headers();
myHeaders.append("x-rapidapi-key", "XxXxXxXxXxXxXxXxXxXxXxXx");
myHeaders.append("x-rapidapi-host", "v1.formula-1.api-sports.io");

var requestOptions = {
  method: 'GET',
  headers: myHeaders,
```

```
  redirect: 'follow'
};

fetch("https://v1.formula-1.api-sports.io/competitions", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

jQuery

```
var settings = {
  "url": "https://v1.formula-1.api-sports.io/competitions",
  "method": "GET",
  "timeout": 0,
  "headers": {
    "x-rapidapi-key": "XxXxXxXxXxXxXxXxXxXxXxXx",
    "x-rapidapi-host": "v1.formula-1.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.formula-1.api-sports.io/competitions");
xhr.setRequestHeader("x-rapidapi-key", "XxXxXxXxXxXxXxXxXxXxXxXx");
xhr.setRequestHeader("x-rapidapi-host", "v1.formula-1.api-sports.io");

xhr.send();
```

NodeJs

Axios

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

var config = {
  method: 'get',
  url: 'https://v1.formula-1.api-sports.io/competitions',
  headers: {
    'x-rapidapi-key': 'XxXxXxXxXxXxXxXxXxXxXxXx',
    'x-rapidapi-host': 'v1.formula-1.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.formula-1.api-sports.io',
  'path': '/competitions',
  'headers': {
    'x-rapidapi-key': 'XxXxXxXxXxXxXxXxXxXxXxXx',
    'x-rapidapi-host': 'v1.formula-1.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.formula-1.api-sports.io/competitions',
  'headers': {
    'x-rapidapi-key': 'XxXxXxXxXxXxXxXxXxXxXxXx',
    'x-rapidapi-host': 'v1.formula-1.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.formula-1.api-sports.io/competitions')
  .headers({
    'x-rapidapi-key': 'XxXxXxXxXxXxXxXxXxXxXxXx',
    'x-rapidapi-host': 'v1.formula-1.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 URLWithS
  cachePolicy:NSURLRequestUseProtocolCachePolicy
  timeoutInterval:10.0];
NSDictionary *headers = @{
  @"x-rapidapi-key": @"XxXxXxXxXxXxXxXxXxXxXxXx",
  @"x-rapidapi-host": @"v1.formula-1.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:dat
    NSLog(@"%@",responseDictionary);
    dispatch_semaphore_signal(sema);
  }
}];
[dataTask resume];
dispatch_semaphore_wait(sema, DISPATCH_TIME_FOREVER);
```

OCaml

Cohttp

```
open Lwt
open Cohttp
open Cohttp_lwt_unix

let reqBody =
  let uri = Uri.of_string "https://v1.formula-1.api-sports.io/competitions" in
  let headers = Header.init ()
    |> fun h -> Header.add h "x-rapidapi-key" "XxXxXxXxXxXxXxXxXxXxXxXx"
    |> fun h -> Header.add h "x-rapidapi-host" "v1.formula-1.api-sports.io"
  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://v1.formula-1.api-sports.io/competitions',
  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-rapidapi-key: XxXxXxXxXxXxXxXxXxXxXxXx',
    'x-rapidapi-host: v1.formula-1.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.formula-1.api-sports.io/competitions');
$request->setMethod(HTTP_Request2::METHOD_GET);
$request->setConfig(array(
  'follow_redirects' => TRUE
));
$request->setHeader(array(
  'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXx',
  'x-rapidapi-host' => 'v1.formula-1.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.formula-1.api-sports.io/competitions');
$request->setRequestMethod('GET');
$request->setHeaders(array(
```

```
    'x-rapidapi-host' => 'v1.formula-1.api-sports.io',
    'x-rapidapi-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-rapidapi-key", "XxXxXxXxXxXxXxXxXxXxXxXx")
$headers.Add("x-rapidapi-host", "v1.formula-1.api-sports.io")

$response = Invoke-RestMethod 'https://v1.formula-1.api-sports.io/competitions' -M
$response | ConvertTo-Json
```

Python

`http.client`

```
import http.client

conn = http.client.HTTPSConnection("v1.formula-1.api-sports.io")

headers = {
    'x-rapidapi-host': "v1.formula-1.api-sports.io",
    'x-rapidapi-key': "XxXxXxXxXxXxXxXxXxXxXxXx"
    }

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

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

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

`Requests`

```
url = "https://v1.formula-1.api-sports.io/competitions"

payload={}
headers = {
  'x-rapidapi-key': 'XxXxXxXxXxXxXxXxXxXxXx',
  'x-rapidapi-host': 'v1.formula-1.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.formula-1.api-sports.io/competitions")

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.formula-1.api-sports.io'
request["x-rapidapi-key"] = 'XxXxXxXxXxXxXxXxXxXxXx'

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

Shell

Httpie

```
http --follow --timeout 3600 GET 'https://v1.formula-1.api-sports.io/competitions'
 x-rapidapi-key:'XxXxXxXxXxXxXxXxXxXxXxXx' \
 x-rapidapi-host:'v1.formula-1.api-sports.io'
```

wget

```
wget --no-check-certificate --quiet \
  --method GET \
  --timeout=0 \
  --header 'x-rapidapi-key: XxXxXxXxXxXxXxXxXxXxXxXx' \
  --header 'x-rapidapi-host: v1.formula-1.api-sports.io' \
   'https://v1.formula-1.api-sports.io/competitions'
```

Swift

URLSession

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

var semaphore = DispatchSemaphore (value: 0)

var request = URLRequest(url: URL(string: "https://v1.formula-1.api-sports.io/comp
request.addValue("XxXxXxXxXxXxXxXxXxXxXxXx", forHTTPHeaderField: "x-rapidapi-key")
request.addValue("v1.formula-1.api-sports.io", forHTTPHeaderField: "x-rapidapi-hos

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

1.3.1

- Add endpoint `rankings/fastestlaps`
- Add endpoint `rankings/startinggrid`
- Add endpoint `pitstops`

1.2.9

- Added a `fastest lap` field in the `races` endpoint which contains the driver id and the time to complete it
- Added more data for the `circuits` endpoint (first_grand_prix, laps, length, race_distance, lap_record)
- Added more data for the `teams` endpoint (base, first_team_entry, world_championships, highest_race_finish, pole_positions, fastest_laps, technical_manager, chassis)
- Added more data for the `drivers` endpoint (abbr, country, birthplace, number, grands_prix_entered, world_championships, podiums, highest_race_finish, highest_grid_position, career_points)
- Update of all teams logos for 2022 season
- Update of all drivers photo for 2022 season

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.

Timezone

timezone

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

> This endpoint does not require any parameters.

HEADER PARAMETERS

`x-rapidapi-key` `required`	string Your API-Key

Responses

> **200** OK

GET /timezone

Request samples

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

Copy

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

$request->setRequestUrl('https://v1.formula-1.api-sports.io/timezone');
$request->setRequestMethod('GET');
$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.formula-1.api-sports.io',
    'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXx'
));

$client->enqueue($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",
        "Africa/Bamako"
    ]
}
```


Seasons

seasons

Get all seasons available.

All seasons are only **4-digit keys**. All results can be used in other endpoints as filters.

> This endpoint does not require any parameters.

HEADER PARAMETERS

`x-rapidapi-key` required	string Your API-Key

Responses

200 OK

GET /seasons

Request samples

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

Copy

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

$request->setRequestUrl('https://v1.formula-1.api-sports.io/seasons');
$request->setRequestMethod('GET');
$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.formula-1.api-sports.io',
    'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXx'
));

$client->enqueue($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": 9,
```

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


Competitions

competitions

Get the list of available competitions.

The competition `id` are **unique** in the API and competitions keep it across all `seasons`

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

QUERY PARAMETERS

`id`	integer The id of the competition
`name`	string Example: `name=Australian Grand Prix` The name of the competition
`country`	string Example: `country=Australia` The name of the country

city	string Example: `city=Melbourne` The name of the city
search	string >= 3 characters Example: `search=Australian` Allow to search for a competition name

HEADER PARAMETERS

x-rapidapi-key required	string Your API-Key

Responses

> **200** OK

GET /competitions

Request samples

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

Copy

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

$request->setRequestUrl('https://v1.formula-1.api-sports.io/competitions');
$request->setRequestMethod('GET');
$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.formula-1.api-sports.io',
    'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXx'
));

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

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

Response samples

200

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "competitions",
  - "parameters": {
        "id": "1"
    },
    "errors": [ ],
    "results": 1,
  - "response": [
      + { … }
    ]
}
```

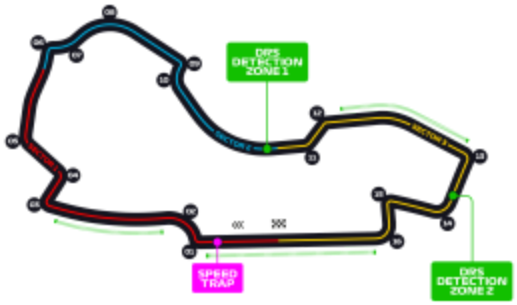
Circuits

circuits

Get the list of available circuits.

The circuit `id` are **unique** in the API and circuits keep it across all `seasons`

Sample `image` of a circuit :



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

QUERY PARAMETERS

`id`	integer The id of the circuit
`competition`	integer The id of the competition
`name`	string Example: `name=Melbourne Grand Prix Circuit` The name of the circuit
`search`	string >= 3 characters Example: `search=Melbourne` Allow to search for a circuit name

HEADER PARAMETERS

`x-rapidapi-key` required	string Your API-Key

Responses

> **200** OK

`GET /circuits`

Request samples

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

Copy

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

$request->setRequestUrl('https://v1.formula-1.api-sports.io/circuits');
$request->setRequestMethod('GET');
$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.formula-1.api-sports.io',
    'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXx'
));

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

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

Response samples

200

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "circuits",
  - "parameters": {
        "id": "1"
    },
    "errors": [ ],
    "results": 1,
  - "response": [
      + { … }
    ]
}
```

Teams

teams

Get the list of available competitions.

The team `id` are **unique** in the API and teams keep it across all `seasons`

Sample `logo` of a team :

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

QUERY PARAMETERS

`id`
integer
The id of the team

`name`
string
Example: `name=Red Bull Racing`
The name of the team

`search`
string >= 3 characters
Example: `search=Red bull`
Allow to search for a team name

HEADER PARAMETERS

`x-rapidapi-key`
`required`
string
Your API-Key

Responses

> **200** OK

GET /teams

Request samples

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

Copy

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

$request->setRequestUrl('https://v1.formula-1.api-sports.io/teams');
$request->setRequestMethod('GET');
$request->setHeaders(array(
    'x-rapidapi-host' => 'v1.formula-1.api-sports.io',
    'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXx'
));

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

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

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
    "get": "teams",
```

```
  - "parameters": {
        "id": "1"
    },
    "errors": [ ],
    "results": 1,
  - "response": [
     + { … }
    ]
}
```


Drivers

drivers

Get the list of available drivers.

The driver `id` are **unique** in the API and drivers keep it across all `seasons`

Sample `image` of a driver :

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

> This endpoint require at least one parameter.

QUERY PARAMETERS

`id`	integer The id of the driver

name	string Example: `name=Lewis Hamilton` The name of the driver
search	string >= 3 characters Example: `search=lewi` Allow to search for a driver name

HEADER PARAMETERS

x-rapidapi-key required	string Your API-Key

Responses

> **200** OK

GET /drivers

Request samples

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

Copy

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

$request->setRequestUrl('https://v1.formula-1.api-sports.io/drivers');
$request->setRequestMethod('GET');
$request->setQuery(new http\QueryString(array(
        'search' => 'lewi'
)));

$request->setHeaders(array(
        'x-rapidapi-host' => 'v1.formula-1.api-sports.io',
        'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXx'
));
```

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

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

Response samples

200

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "drivers",
  - "parameters": {
        "search": "lewi"
    },
    "errors": [ ],
    "results": 1,
  - "response": [
      + { … }
    ]
}
```

Races

races

Get the list of available **races** for a competition.

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

- Live
- Completed
- Cancelled
- Postponed
- Scheduled

Available Types

- Race
- 1st Qualifying
- 2nd Qualifying
- 3rd Qualifying
- Sprint
- 1st Sprint Shootout
- 2nd Sprint Shootout
- 3rd Sprint Shootout
- 1st Practice
- 2nd Practice
- 3rd Practice

> This endpoint requires at least one of these parameters `id`, `date`, `next`, `last` and `season`.

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

QUERY PARAMETERS

`id`
integer
The id of the race

`date`
stringYYYY-MM-DD
Example: `date=2021-12-12`
A valid date

`next`
integer `<= 2 characters`
Example: `next=30`
The x next races

`last`
integer `<= 2 characters`
Example: `last=30`
The x last races

`competition`
integer
The id of the competition

circuit	integer The id of the circuit
season	integer = 4 characters YYYY The season of the race
type	string Enum: "Race" "1st Qualifying" "2nd Qualifying" "3rd Qualifying" "Sprint" "1st Sprint Shootout" "2nd Sprint Shootout" "3rd Sprint Shootout" "1st Practice" "2nd Practice" "3rd Practice" The type of the race
timezone	string Example: timezone=Europe/London A valid timezone

HEADER PARAMETERS

x-rapidapi-key required	string Your API-Key

Responses

> **200** OK

GET /races

Request samples

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

Copy

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

$request->setRequestUrl('https://v1.formula-1.api-sports.io/races');
```

```
$request->setRequestMethod('GET');
$request->setQuery(new http\QueryString(array(
        'competition' => '23',
        'season' => '2021'
)));

$request->setHeaders(array(
        'x-rapidapi-host' => 'v1.formula-1.api-sports.io',
        'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXx'
));

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

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

Response samples

200

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "races",
  - "parameters": {
        "season": "2021",
        "type": "race",
        "competition": "23"
    },
    "errors": [ ],
    "results": 1,
  - "response": [
      + { … }
    ]
}
```

Rankings

rankings/teams

Get the teams rankings for a season.

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

QUERY PARAMETERS

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

HEADER PARAMETERS

`x-rapidapi-key` required	string Your API-Key

Responses

> **200** OK

`GET /rankings/teams`

Request samples

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

Copy

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

$request->setRequestUrl('https://v1.formula-1.api-sports.io/rankings/teams');
$request->setRequestMethod('GET');
$request->setQuery(new http\QueryString(array(
        'season' => '2019'
)));

$request->setHeaders(array(
        'x-rapidapi-host' => 'v1.formula-1.api-sports.io',
        'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXx'
));

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

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

Response samples

200

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "rankings",
  - "parameters": {
        "season": "2019"
    },
    "errors": [ ],
    "results": 10,
```

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


---

rankings/drivers

Get the drivers rankings for a season.

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

QUERY PARAMETERS

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

HEADER PARAMETERS

`x-rapidapi-key` required	string Your API-Key

Responses

> **200** OK

GET /rankings/drivers

Request samples

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

Copy

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

$request->setRequestUrl('https://v1.formula-1.api-sports.io/rankings/drivers');
$request->setRequestMethod('GET');
$request->setQuery(new http\QueryString(array(
        'season' => '2019'
)));

$request->setHeaders(array(
        'x-rapidapi-host' => 'v1.formula-1.api-sports.io',
        'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXx'
));

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

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

Response samples

200

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "rankings",
```

```
- "parameters": {
      "season": "2019"
  },
  "errors": [ ],
  "results": 20,
- "response": [
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … },
    + { … }
  ]
}
```


rankings/races

Get the rankings for a race.

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

QUERY PARAMETERS

`race` required	integer The id of the race

team	integer The id of the team
driver	integer The id of the driver

HEADER PARAMETERS

x-rapidapi-key required	string Your API-Key

Responses

> **200** OK

GET /rankings/races

Request samples

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

Copy

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

$request->setRequestUrl('https://v1.formula-1.api-sports.io/rankings/races');
$request->setRequestMethod('GET');
$request->setQuery(new http\QueryString(array(
        'race' => '50'
)));

$request->setHeaders(array(
        'x-rapidapi-host' => 'v1.formula-1.api-sports.io',
        'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXx'
));

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

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

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

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
    "get": "rankings",
  - "parameters": {
        "race": "50"
    },
    "errors": [ ],
    "results": 20,
  - "response": [
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … }
    ]
}
```


---

rankings/fastestlaps

Get the ranking of the fastest laps for a race.

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

QUERY PARAMETERS

race required	integer The id of the race
team	integer The id of the team
driver	integer The id of the driver

HEADER PARAMETERS

x-rapidapi-key required	string Your API-Key

Responses

> **200** OK

GET /rankings/fastestlaps

Request samples

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

Copy

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

$request->setRequestUrl('https://v1.formula-1.api-sports.io/rankings/fastestlaps
$request->setRequestMethod('GET');
$request->setQuery(new http\QueryString(array(
        'race' => '50'
)));

$request->setHeaders(array(
        'x-rapidapi-host' => 'v1.formula-1.api-sports.io',
        'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXx'
```

```
));

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

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

Response samples

200

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "rankings",
  - "parameters": {
        "race": "50"
    },
    "errors": [ ],
    "results": 20,
```

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


---

rankings/startinggrid

Get the starting grid for a race.

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

QUERY PARAMETERS

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

driver	integer The id of the driver

HEADER PARAMETERS

x-rapidapi-key required	string Your API-Key

Responses

> **200** OK

GET /rankings/startinggrid

Request samples

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

Copy

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

$request->setRequestUrl('https://v1.formula-1.api-sports.io/rankings/startinggri
$request->setRequestMethod('GET');
$request->setQuery(new http\QueryString(array(
        'race' => '50'
)));

$request->setHeaders(array(
        'x-rapidapi-host' => 'v1.formula-1.api-sports.io',
        'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXx'
));

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

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

Response samples

200

Content type
application/json

Copy Expand all Collapse all

```
{
    "get": "rankings",
  - "parameters": {
        "race": "50"
    },
    "errors": [ ],
    "results": 20,
  - "response": [
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … },
      + { … }
    ]
}
```

Pit Stops

pitstops

Get the list of pit stops made by all drivers during a race.

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

QUERY PARAMETERS

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

HEADER PARAMETERS

`x-rapidapi-key` required	string Your API-Key

Responses

> **200** OK

`GET /pitstops`

Request samples

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

Copy

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

$request->setRequestUrl('https://v1.formula-1.api-sports.io/pitstops');
$request->setRequestMethod('GET');
$request->setQuery(new http\QueryString(array(
```

```
        'race' => '50'
]));

$request->setHeaders(array(
        'x-rapidapi-host' => 'v1.formula-1.api-sports.io',
        'x-rapidapi-key' => 'XxXxXxXxXxXxXxXxXxXxXxXx'
));

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

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

Response samples

200

Content type
application/json

Copy Expand all Collapse all


```
{
    "get": "pitstop",
  - "parameters": {
        "race": "50"
    },
    "errors": [ ],
    "results": 22,
```

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