```
SOCCERDATAAPI DOCS DUMP
================================================================================
Source: https://soccerdataapi.com/docs/

GLOBAL API URLS FOUND ON PAGE
--------------------------------------------------------------------------------
- https://api.soccerdataapi.com/country/?auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/country/
- https://api.soccerdataapi.com/league/?country_id=1&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/league/?country_id=1auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/league/
- https://api.soccerdataapi.com/season/?league_id=228&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/season/
- https://api.soccerdataapi.com/stage/?league_id=310&season=2022-
2023&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/stage/
- https://api.soccerdataapi.com/group/?stage_id=8646&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/group/
- https://api.soccerdataapi.com/stadium/?team_id=4138&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/stadium/
- https://api.soccerdataapi.com/team/?team_id=4138&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/team/
- https://api.soccerdataapi.com/player/?player_id=61793&auth_token=YOUR-AUTH-
TOKEN
- https://api.soccerdataapi.com/player/
- https://api.soccerdataapi.com/transfers/?team_id=4138&auth_token=YOUR-AUTH-
TOKEN
- https://api.soccerdataapi.com/transfers/
- https://api.soccerdataapi.com/head-to-head/?
team_1_id=4137&team_2_id=4149&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/head-to-head/
- https://api.soccerdataapi.com/standing/?league_id=228&auth_token=YOUR-AUTH-
TOKEN
- https://api.soccerdataapi.com/standing/
- https://api.soccerdataapi.com/livescores/?auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/livescores/
- https://api.soccerdataapi.com/matches/?league_id=228&auth_token=YOUR-AUTH-
TOKEN
- https://api.soccerdataapi.com/matches/
- https://api.soccerdataapi.com/match/?match_id=531585&auth_token=YOUR-AUTH-
TOKEN
- https://api.soccerdataapi.com/match/
- https://api.soccerdataapi.com/match-preview/?match_id=544770&auth_token=YOUR-
AUTH-TOKEN
- https://api.soccerdataapi.com/match-preview/
- https://api.soccerdataapi.com/match-previews-upcoming/?auth_token=YOUR-AUTH-
TOKEN
- https://api.soccerdataapi.com/match-previews-upcoming/

[1] Get Started
Anchor ID: content-get-started
--------------------------------------------------------------------------------
CODE / EXAMPLES / RESPONSES (raw code blocks)
--- CODE BLOCK 1 ---
API Endpoint
 https://api.soccerdataapi.com
--- END CODE BLOCK 1 ---

NARRATIVE TEXT (section plain text)
Get Started
API Endpoint
 https://api.soccerdataapi.com
The Soccerdata API provides live scores, league stats and in-depth pre-match
content for 125+ worldwide leagues.
```

Data types include live scores, league stats, transfers, injuries, head-to-head stats, white-label odds, A.I. powered match previews, projected and live team lineups, weather forecasts and game winner and over/under predictions.
To access the API endpoints,
sign-up for an account
and obtain an
API key
.
An
API key
should be included in all requests using
auth_token
as a parameter:
api.soccerdataapi.com/livescores/?auth_token=YOUR-AUTH-TOKEN
JSON data is returned in gzip compressed format. Every API call must include the
{'Accept-Encoding': 'gzip'}
request header or it will fail.
More Info on Accept-Encoding headers.

==

[2] Get Country
Anchor ID: content-get-country

ENDPOINT URLS
- https://api.soccerdataapi.com/country/?auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/country/

CODE / EXAMPLES / RESPONSES (raw code blocks)

--- CODE BLOCK 1 ---

```
# Get Country: Javascript Example Request
async function getCountries() {
 const response = await fetch("https://api.soccerdataapi.com/country/?auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

--- END CODE BLOCK 1 ---

--- CODE BLOCK 2 ---

```
# Get Country: Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/country/?auth_token=YOUR-AUTH-TOKEN'
```

--- END CODE BLOCK 2 ---

--- CODE BLOCK 3 ---

```
# Get Country: Python Example Request
import requests
url = "https://api.soccerdataapi.com/country/"
querystring = {'auth_token': YOUR-AUTH-TOKEN}
```

```
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
--- END CODE BLOCK 3 ---
```

```
--- CODE BLOCK 4 ---
Get Country: Example JSON Response
{
 "count": 221,
 "next": null,
 "previous": null,
 "results": [
 {
 "id": 201,
 "name": "afghanistan"
 },
 {
 "id": 47,
 "name": "albania"
 },
 {
 "id": 87,
 "name": "algeria"
 },
 {
 "id": 224,
 "name": "american samoa"
 },
 {
 "id": 55,
 "name": "andorra"
 },
 ...
 ]
}
--- END CODE BLOCK 4 ---
```

NARRATIVE TEXT (section plain text)
Get Country

```
# Get Country: Javascript Example Request
async function getCountries() {
 const response = await fetch("https://api.soccerdataapi.com/country/?
auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
# Get Country: Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/country/?auth_token=YOUR-AUTH-TOKEN'
# Get Country: Python Example Request
import requests
url = "https://api.soccerdataapi.com/country/"
querystring = {'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
Retrieve a list of countries with a GET request to the endpoint:
https://api.soccerdataapi.com/country/
Get Country: Example JSON Response
{
 "count": 221,
 "next": null,
 "previous": null,
 "results": [
 {
 "id": 201,
 "name": "afghanistan"
 },
 {
 "id": 47,
 "name": "albania"
 },
 {
 "id": 87,
 "name": "algeria"
 },
 {
 "id": 224,
 "name": "american samoa"
 },
 {
 "id": 55,
 "name": "andorra"
 },
 ...
 ]
}

================================================================================

[3] Get League
Anchor ID: content-get-league
--------------------------------------------------------------------------------
ENDPOINT URLS
- https://api.soccerdataapi.com/league/?country_id=1&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/league/?country_id=1auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/league/

QUERY PARAMETERS (table rows)
- country_id | Integer | (optional) Get leagues by country_id

CODE / EXAMPLES / RESPONSES (raw code blocks)
--- CODE BLOCK 1 ---
# Get League: Javascript Example Request
async function getLeagues() {
 const response = await fetch("https://api.soccerdataapi.com/league/?
country_id=1&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
--- END CODE BLOCK 1 ---
```

```
--- CODE BLOCK 2 ---
# Get League: Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'
 --url 'https://api.soccerdataapi.com/league/?country_id=1auth_token=YOUR-AUTH-
TOKEN'
--- END CODE BLOCK 2 ---
```

```
--- CODE BLOCK 3 ---
# Get League: Python Example Request
import requests
url = "https://api.soccerdataapi.com/league/"
querystring = {'country_id': 1, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
--- END CODE BLOCK 3 ---
```

```
--- CODE BLOCK 4 ---
Get League: Example JSON Response
{
 "count": 129,
 "next": null,
 "previous": null,
 "results": [
 {
 "id": 166,
 "country": {
 "id": 1,
 "name": "usa"
 },
 "name": "USL Championship",
 "is_cup": false
 },
 {
 "id": 168,
 "country": {
 "id": 1,
 "name": "usa"
 },
 "name": "MLS",
 "is_cup": false
 },
 {
```

```
 "id": 197,
 "country": {
 "id": 2,
 "name": "canada"
 },
 "name": "Canadian Premier League",
 "is_cup": false
 },
 {
 "id": 198,
 "country": {
 "id": 4,
 "name": "europe"
 },
 "name": "Europa Conference League",
 "is_cup": true
 },
 ...
 ]
}
```

--- END CODE BLOCK 4 ---

NARRATIVE TEXT (section plain text)

Get League

# Get League: Javascript Example Request

```
async function getLeagues() {
 const response = await fetch("https://api.soccerdataapi.com/league/?country_id=1&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

# Get League: Curl Example Request

```
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'
 --url 'https://api.soccerdataapi.com/league/?country_id=1auth_token=YOUR-AUTH-TOKEN'
```

# Get League: Python Example Request

```
import requests
url = "https://api.soccerdataapi.com/league/"
querystring = {'country_id': 1, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

Retrieve a list of leagues with a GET request to the endpoint:
https://api.soccerdataapi.com/league/

Get League: Example JSON Response

```
{
 "count": 129,
 "next": null,
 "previous": null,
 "results": [
 {
 "id": 166,
 "country": {
 "id": 1,
 "name": "usa"
 },
 "name": "USL Championship",
 "is_cup": false
 },
 {
 "id": 168,
 "country": {
 "id": 1,
 "name": "usa"
 },
 "name": "MLS",
 "is_cup": false
 },
 {
 "id": 197,
 "country": {
 "id": 2,
 "name": "canada"
 },
 "name": "Canadian Premier League",
 "is_cup": false
 },
 {
 "id": 198,
 "country": {
 "id": 4,
 "name": "europe"
 },
 "name": "Europa Conference League",
 "is_cup": true
 },
 ...
 ]
}
QUERY PARAMETERS
Field
Type
Description
country_id
Integer
(optional) Get leagues by country_id

================================================================================

[4] Get Season
Anchor ID: content-get-season
--------------------------------------------------------------------------------
ENDPOINT URLS
- https://api.soccerdataapi.com/season/?league_id=228&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/season/

QUERY PARAMETERS (table rows)
- league_id | Integer | (required) Get seasons by league_id

CODE / EXAMPLES / RESPONSES (raw code blocks)
--- CODE BLOCK 1 ---
```

```
# Get Season: Javascript Example Request
async function getSeasons() {
 const response = await fetch("https://api.soccerdataapi.com/season/?
league_id=228&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

--- END CODE BLOCK 1 ---

--- CODE BLOCK 2 ---

```
# Get Season: Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/season/?league_id=228&auth_token=YOUR-AUTH-TOKEN'
```

--- END CODE BLOCK 2 ---

--- CODE BLOCK 3 ---

```
# Get Season: Python Example Request
import requests
url = "https://api.soccerdataapi.com/season/"
querystring = {'league_id': 228, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

--- END CODE BLOCK 3 ---

--- CODE BLOCK 4 ---

Get Season: Example JSON Response

```
{
 "count": 7,
 "next": null,
 "previous": null,
 "results": [
 {
 "id": 4354,
 "league": {
 "id": 228,
 "name": "Premier League"
 },
 "year": "2023-2024",
 "is_active": true
 },
 {
 "id": 3807,
 "league": {
 "id": 228,
 "name": "Premier League"
 },
 "year": "2022-2023",
 "is_active": false
 },
 {
 "id": 3806,
 "league": {
 "id": 228,
 "name": "Premier League"
 },
 "year": "2021-2022",
 "is_active": false
 },
 {
 "id": 3805,
 "league": {
 "id": 228,
 "name": "Premier League"
 },
 "year": "2020-2021",
 "is_active": false
 },
 {
 "id": 3804,
 "league": {
 "id": 228,
 "name": "Premier League"
 },
 "year": "2019-2020",
 "is_active": false
 },
 {
 "id": 3803,
 "league": {
 "id": 228,
 "name": "Premier League"
 },
 "year": "2018-2019",
 "is_active": false
 },
 {
 "id": 3802,
 "league": {
 "id": 228,
 "name": "Premier League"
 },
 "year": "2017-2018",
 "is_active": false
 }
 ]
}
```

--- END CODE BLOCK 4 ---

NARRATIVE TEXT (section plain text)

Get Season

# Get Season: Javascript Example Request

```
async function getSeasons() {
 const response = await fetch("https://api.soccerdataapi.com/season/?
league_id=228&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

# Get Season: Curl Example Request

```
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/season/?league_id=228&auth_token=YOUR-AUTH-TOKEN'
```

# Get Season: Python Example Request

```
import requests
url = "https://api.soccerdataapi.com/season/"
querystring = {'league_id': 228, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

Retrieve a list of seasons for league with a GET request to the endpoint: https://api.soccerdataapi.com/season/

Get Season: Example JSON Response

```
{
 "count": 7,
 "next": null,
 "previous": null,
 "results": [
 {
 "id": 4354,
 "league": {
 "id": 228,
 "name": "Premier League"
 },
 "year": "2023-2024",
 "is_active": true
 },
 {
 "id": 3807,
 "league": {
 "id": 228,
 "name": "Premier League"
 },
 "year": "2022-2023",
 "is_active": false
 },
 {
 "id": 3806,
 "league": {
 "id": 228,
 "name": "Premier League"
 },
 "year": "2021-2022",
 "is_active": false
 },
 {
 "id": 3805,
 "league": {
 "id": 228,
 "name": "Premier League"
 },
 "year": "2020-2021",
 "is_active": false
 },
 {
 "id": 3804,
 "league": {
 "id": 228,
 "name": "Premier League"
 },
 "year": "2019-2020",
 "is_active": false
 },
 {
 "id": 3803,
 "league": {
 "id": 228,
 "name": "Premier League"
 },
 "year": "2018-2019",
 "is_active": false
 },
 {
 "id": 3802,
 "league": {
 "id": 228,
 "name": "Premier League"
 },
 "year": "2017-2018",
 "is_active": false
 }
 ]
}
```

QUERY PARAMETERS
Field
Type
Description
league_id
Integer
(required) Get seasons by league_id

==

[5] Get Season Stages
Anchor ID: content-get-season-stage

ENDPOINT URLS
- https://api.soccerdataapi.com/stage/?league_id=310&season=2022-2023&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/stage/

QUERY PARAMETERS (table rows)
- league_id | Integer | (required) Get stages by league_id. Defaults to current season.
- season | String | (optional) Get stages by league_id and season

CODE / EXAMPLES / RESPONSES (raw code blocks)
--- CODE BLOCK 1 ---

```
# Get Season Stages: Javascript Example Request
async function getSeasonStages() {
 const response = await fetch("https://api.soccerdataapi.com/stage/?
league_id=310&season=2022-2023&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
```

```
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
--- END CODE BLOCK 1 ---

--- CODE BLOCK 2 ---
# Get Season Stages: Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/stage/?league_id=310&season=2022-
2023&auth_token=YOUR-AUTH-TOKEN'
--- END CODE BLOCK 2 ---

--- CODE BLOCK 3 ---
# Get Season Stages: Python Example Request
import requests
url = "https://api.soccerdataapi.com/stage/"
querystring = {'league_id': 310, 'season': '2022-2023', 'auth_token': YOUR-AUTH-
TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
--- END CODE BLOCK 3 ---

--- CODE BLOCK 4 ---
Get Season Stage: Example JSON Response
{
 "count": 11,
 "next": null,
 "previous": null,
 "results": [
 {
 "id": 8667,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "season": "2022-2023",
 "name": "Preliminary Round - Semi-finals",
 "has_groups": false,
 "is_active": false
 },
 {
 "id": 8666,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "season": "2022-2023",
 "name": "Preliminary Round - Final",
"has_groups": false,
"is_active": false
},
{
"id": 8662,
"league": {
"id": 310,
"name": "UEFA Champions League"
},
"season": "2022-2023",
"name": "1st Qualifying Round",
"has_groups": false,
"is_active": false
},
{
"id": 8661,
"league": {
"id": 310,
"name": "UEFA Champions League"
},
"season": "2022-2023",
"name": "2nd Qualifying Round",
"has_groups": false,
"is_active": false
},
{
"id": 8659,
"league": {
"id": 310,
"name": "UEFA Champions League"
},
"season": "2022-2023",
"name": "3rd Qualifying Round",
"has_groups": false,
"is_active": false
},
{
"id": 8658,
"league": {
"id": 310,
"name": "UEFA Champions League"
},
"season": "2022-2023",
"name": "Play-offs",
"has_groups": false,
"is_active": false
},
{
"id": 8646,
"league": {
"id": 310,
"name": "UEFA Champions League"
},
"season": "2022-2023",
"name": "Group Stage",
"has_groups": true,
"is_active": false
},
{
"id": 8645,
"league": {
"id": 310,
"name": "UEFA Champions League"
},
 "season": "2022-2023",
 "name": "Round of 16",
 "has_groups": false,
 "is_active": false
 },
 {
 "id": 8644,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "season": "2022-2023",
 "name": "Quarter-finals",
 "has_groups": false,
 "is_active": false
 }
 {
 "id": 8643,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "season": "2022-2023",
 "name": "Semi-finals",
 "has_groups": false,
 "is_active": false
 },
 {
 "id": 8642,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "season": "2022-2023",
 "name": "Final",
 "has_groups": false,
 "is_active": false
 }
 ]
}
```

--- END CODE BLOCK 4 ---

NARRATIVE TEXT (section plain text)

Get Season Stages

# Get Season Stages: Javascript Example Request

```
async function getSeasonStages() {
 const response = await fetch("https://api.soccerdataapi.com/stage/?league_id=310&season=2022-2023&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

# Get Season Stages: Curl Example Request

```
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/stage/?league_id=310&season=2022-
2023&auth_token=YOUR-AUTH-TOKEN'
# Get Season Stages: Python Example Request
import requests
url = "https://api.soccerdataapi.com/stage/"
querystring = {'league_id': 310, 'season': '2022-2023', 'auth_token': YOUR-AUTH-
TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

Retrieve a list of stages for league season with a GET request to the endpoint: https://api.soccerdataapi.com/stage/

Get Season Stage: Example JSON Response

```
{
 "count": 11,
 "next": null,
 "previous": null,
 "results": [
 {
 "id": 8667,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "season": "2022-2023",
 "name": "Preliminary Round - Semi-finals",
 "has_groups": false,
 "is_active": false
 },
 {
 "id": 8666,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "season": "2022-2023",
 "name": "Preliminary Round - Final",
 "has_groups": false,
 "is_active": false
 },
 {
 "id": 8662,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "season": "2022-2023",
 "name": "1st Qualifying Round",
 "has_groups": false,
 "is_active": false
 },
 {
 "id": 8661,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "season": "2022-2023",
"name": "2nd Qualifying Round",
"has_groups": false,
"is_active": false
},
{
"id": 8659,
"league": {
"id": 310,
"name": "UEFA Champions League"
},
"season": "2022-2023",
"name": "3rd Qualifying Round",
"has_groups": false,
"is_active": false
},
{
"id": 8658,
"league": {
"id": 310,
"name": "UEFA Champions League"
},
"season": "2022-2023",
"name": "Play-offs",
"has_groups": false,
"is_active": false
},
{
"id": 8646,
"league": {
"id": 310,
"name": "UEFA Champions League"
},
"season": "2022-2023",
"name": "Group Stage",
"has_groups": true,
"is_active": false
},
{
"id": 8645,
"league": {
"id": 310,
"name": "UEFA Champions League"
},
"season": "2022-2023",
"name": "Round of 16",
"has_groups": false,
"is_active": false
},
{
"id": 8644,
"league": {
"id": 310,
"name": "UEFA Champions League"
},
"season": "2022-2023",
"name": "Quarter-finals",
"has_groups": false,
"is_active": false
}
{
"id": 8643,
"league": {
"id": 310,
"name": "UEFA Champions League"
```

```
 },
 "season": "2022-2023",
 "name": "Semi-finals",
 "has_groups": false,
 "is_active": false
 },
 {
 "id": 8642,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "season": "2022-2023",
 "name": "Final",
 "has_groups": false,
 "is_active": false
 }
 ]
}
```

QUERY PARAMETERS
Field
Type
Description
league_id
Integer
(required) Get stages by league_id. Defaults to current season.
season
String
(optional) Get stages by league_id and season

==

[6] Get Groups
Anchor ID: content-get-group

ENDPOINT URLS
- https://api.soccerdataapi.com/group/?stage_id=8646&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/group/

QUERY PARAMETERS (table rows)
- stage_id | Integer | (required) Get groups by stage_id

CODE / EXAMPLES / RESPONSES (raw code blocks)
--- CODE BLOCK 1 ---

```
# Get Groups: Javascript Example Request
async function getGroup() {
 const response = await fetch("https://api.soccerdataapi.com/group/?
stage_id=8646&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

--- END CODE BLOCK 1 ---

```
--- CODE BLOCK 2 ---
# Get Groups: Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/group/?stage_id=8646&auth_token=YOUR-AUTH-TOKEN'
--- END CODE BLOCK 2 ---

--- CODE BLOCK 3 ---
# Get Groups: Python Example Request
import requests
url = "https://api.soccerdataapi.com/group/"
querystring = {'stage_id': 8646, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
--- END CODE BLOCK 3 ---

--- CODE BLOCK 4 ---
Get Group: Example JSON Response
{
 "count": 8,
 "next": null,
 "previous": null,
 "results": [
 {
 "id": 689,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "stage": {
 "id": 8646,
 "name": "Group Stage"
 },
 "name": "Group A"
 },
 {
 "id": 690,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "stage": {
 "id": 8646,
 "name": "Group Stage"
 },
 "name": "Group B"
 },
 {
 "id": 691,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "stage": {
 "id": 8646,
 "name": "Group Stage"
 },
 "name": "Group C"
 },
 {
 "id": 692,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "stage": {
 "id": 8646,
 "name": "Group Stage"
 },
 "name": "Group D"
 },
 {
 "id": 693,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "stage": {
 "id": 8646,
 "name": "Group Stage"
 },
 "name": "Group E"
 },
 {
 "id": 694,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "stage": {
 "id": 8646,
 "name": "Group Stage"
 },
 "name": "Group F"
 },
 {
 "id": 695,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "stage": {
 "id": 8646,
 "name": "Group Stage"
 },
 "name": "Group G"
 },
 {
 "id": 696,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "stage": {
 "id": 8646,
 "name": "Group Stage"
 },
 "name": "Group H"
 }
 ]
}
```

--- END CODE BLOCK 4 ---

NARRATIVE TEXT (section plain text)

Get Groups

# Get Groups: Javascript Example Request

```
async function getGroup() {
 const response = await fetch("https://api.soccerdataapi.com/group/?
stage_id=8646&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

# Get Groups: Curl Example Request

```
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/group/?stage_id=8646&auth_token=YOUR-AUTH-TOKEN'
```

# Get Groups: Python Example Request

```
import requests
url = "https://api.soccerdataapi.com/group/"
querystring = {'stage_id': 8646, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

Retrieve a list of groups for stage with a GET request to the endpoint:
https://api.soccerdataapi.com/group/

Get Group: Example JSON Response

```
{
 "count": 8,
 "next": null,
 "previous": null,
 "results": [
 {
 "id": 689,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "stage": {
 "id": 8646,
 "name": "Group Stage"
 },
 "name": "Group A"
 },
 {
 "id": 690,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "stage": {
```

```
"id": 8646,
"name": "Group Stage"
},
"name": "Group B"
},
{
"id": 691,
"league": {
"id": 310,
"name": "UEFA Champions League"
},
"stage": {
"id": 8646,
"name": "Group Stage"
},
"name": "Group C"
},
{
"id": 692,
"league": {
"id": 310,
"name": "UEFA Champions League"
},
"stage": {
"id": 8646,
"name": "Group Stage"
},
"name": "Group D"
},
{
"id": 693,
"league": {
"id": 310,
"name": "UEFA Champions League"
},
"stage": {
"id": 8646,
"name": "Group Stage"
},
"name": "Group E"
},
{
"id": 694,
"league": {
"id": 310,
"name": "UEFA Champions League"
},
"stage": {
"id": 8646,
"name": "Group Stage"
},
"name": "Group F"
},
{
"id": 695,
"league": {
"id": 310,
"name": "UEFA Champions League"
},
"stage": {
"id": 8646,
"name": "Group Stage"
},
"name": "Group G"
```

```
 },
 {
 "id": 696,
 "league": {
 "id": 310,
 "name": "UEFA Champions League"
 },
 "stage": {
 "id": 8646,
 "name": "Group Stage"
 },
 "name": "Group H"
 }
 ]
}
```

QUERY PARAMETERS
Field
Type
Description
stage_id
Integer
(required) Get groups by stage_id

==

[7] Get Stadium
Anchor ID: content-get-stadium

ENDPOINT URLS
- https://api.soccerdataapi.com/stadium/?team_id=4138&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/stadium/

QUERY PARAMETERS (table rows)
- stadium_id | Integer | (optionally required) Get stadium by stadium_id
- team_id | Integer | (optionally required) Get stadium by team_id

CODE / EXAMPLES / RESPONSES (raw code blocks)
--- CODE BLOCK 1 ---

```
# Get Stadium: Javascript Example Request
async function getStadium() {
 const response = await fetch("https://api.soccerdataapi.com/stadium/?
team_id=4138&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

--- END CODE BLOCK 1 ---

--- CODE BLOCK 2 ---

```
# Get Stadium: Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
```

```
'https://api.soccerdataapi.com/stadium/?team_id=4138&auth_token=YOUR-AUTH-TOKEN'
```

--- END CODE BLOCK 2 ---

--- CODE BLOCK 3 ---

```
# Get Stadium: Python Example Request
import requests
url = "https://api.soccerdataapi.com/stadium/"
querystring = {'team_id': 4138, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

--- END CODE BLOCK 3 ---

--- CODE BLOCK 4 ---

Get Stadium: Example JSON Response

```
{
 "id": 2075,
 "teams": [
 {
 "id": 4138,
 "name": "Liverpool"
 }
 ],
 "name": "Anfield",
 "city": "Liverpool"
}
```

--- END CODE BLOCK 4 ---

NARRATIVE TEXT (section plain text)

Get Stadium

```
# Get Stadium: Javascript Example Request
async function getStadium() {
 const response = await fetch("https://api.soccerdataapi.com/stadium/?
team_id=4138&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
# Get Stadium: Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/stadium/?team_id=4138&auth_token=YOUR-AUTH-TOKEN'
# Get Stadium: Python Example Request
import requests
url = "https://api.soccerdataapi.com/stadium/"
querystring = {'team_id': 4138, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

Retrieve stadium by team or stadium id with a GET request to the endpoint:
https://api.soccerdataapi.com/stadium/

Requires either team_id or stadium_id parameters.

Get Stadium: Example JSON Response

```
{
 "id": 2075,
 "teams": [
 {
 "id": 4138,
 "name": "Liverpool"
 }
 ],
 "name": "Anfield",
 "city": "Liverpool"
}
```

QUERY PARAMETERS

Field	Type	Description
stadium_id	Integer	(optionally required) Get stadium by stadium_id
team_id	Integer	(optionally required) Get stadium by team_id

==

[8] Get Team
Anchor ID: content-get-team

ENDPOINT URLS
- https://api.soccerdataapi.com/team/?team_id=4138&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/team/

QUERY PARAMETERS (table rows)
- team_id | Integer | (required) Get team by team_id

CODE / EXAMPLES / RESPONSES (raw code blocks)

```
--- CODE BLOCK 1 ---
# Get Team: Javascript Example Request
async function getTeam() {
 const response = await fetch("https://api.soccerdataapi.com/team/?
team_id=4138&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
--- END CODE BLOCK 1 ---
```

--- CODE BLOCK 2 ---

```
# Get Team: Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/team/?team_id=4138&auth_token=YOUR-AUTH-TOKEN'
```

--- END CODE BLOCK 2 ---

--- CODE BLOCK 3 ---

```
# Get Team: Python Example Request
import requests
url = "https://api.soccerdataapi.com/team/"
querystring = {'team_id': 4138, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

--- END CODE BLOCK 3 ---

--- CODE BLOCK 4 ---

Get Team: Example JSON Response

```
{
 "id": 4138,
 "name": "Liverpool",
 "country": {
 "id": 8,
 "name": "england"
 },
 "stadium": {
 "id": 2075,
 "name": "Anfield",
 "city": "Liverpool"
 },
 "is_nation": false
}
```

--- END CODE BLOCK 4 ---

NARRATIVE TEXT (section plain text)

Get Team

```
# Get Team: Javascript Example Request
async function getTeam() {
 const response = await fetch("https://api.soccerdataapi.com/team/?
team_id=4138&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
# Get Team: Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
```

```
'https://api.soccerdataapi.com/team/?team_id=4138&auth_token=YOUR-AUTH-TOKEN'
# Get Team: Python Example Request
import requests
url = "https://api.soccerdataapi.com/team/"
querystring = {'team_id': 4138, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

Retrieve team by id with a GET request to the endpoint:
https://api.soccerdataapi.com/team/

Get Team: Example JSON Response

```
{
 "id": 4138,
 "name": "Liverpool",
 "country": {
 "id": 8,
 "name": "england"
 },
 "stadium": {
 "id": 2075,
 "name": "Anfield",
 "city": "Liverpool"
 },
 "is_nation": false
}
```

QUERY PARAMETERS
Field
Type
Description
team_id
Integer
(required) Get team by team_id

==

[9] Get Player
Anchor ID: content-get-player

ENDPOINT URLS
- https://api.soccerdataapi.com/player/?player_id=61793&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/player/

QUERY PARAMETERS (table rows)
- player_id | Integer | (required) Get player by player_id

CODE / EXAMPLES / RESPONSES (raw code blocks)
--- CODE BLOCK 1 ---

```
# Get Player: Javascript Example Request
async function getPlayer() {
 const response = await fetch("https://api.soccerdataapi.com/player/?player_id=61793&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

--- END CODE BLOCK 1 ---

--- CODE BLOCK 2 ---

```
# Get Player: Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/player/?player_id=61793&auth_token=YOUR-AUTH-
TOKEN'
```

--- END CODE BLOCK 2 ---

--- CODE BLOCK 3 ---

```
# Get Player: Python Example Request
import requests
url = "https://api.soccerdataapi.com/player/"
querystring = {'player_id': 61793, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

--- END CODE BLOCK 3 ---

--- CODE BLOCK 4 ---

Get Player: Example JSON Response

```
{
 "id": 61793,
 "name": "J. Henderson",
 "team": {
 "id": 4138,
 "name": "Liverpool"
 }
}
```

--- END CODE BLOCK 4 ---

NARRATIVE TEXT (section plain text)

Get Player

```
# Get Player: Javascript Example Request
async function getPlayer() {
 const response = await fetch("https://api.soccerdataapi.com/player/?
player_id=61793&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
# Get Player: Curl Example Request
```

```
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/player/?player_id=61793&auth_token=YOUR-AUTH-
TOKEN'
# Get Player: Python Example Request
import requests
url = "https://api.soccerdataapi.com/player/"
querystring = {'player_id': 61793, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

Retrieve player by id with a GET request to the endpoint:
https://api.soccerdataapi.com/player/
Get Player: Example JSON Response

```
{
 "id": 61793,
 "name": "J. Henderson",
 "team": {
 "id": 4138,
 "name": "Liverpool"
 }
}
```

QUERY PARAMETERS
Field
Type
Description
player_id
Integer
(required) Get player by player_id

==

[10] Get Transfers
Anchor ID: content-get-transfers

ENDPOINT URLS
- https://api.soccerdataapi.com/transfers/?team_id=4138&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/transfers/

QUERY PARAMETERS (table rows)
- team_id | Integer | (required) Get transfers by team_id

CODE / EXAMPLES / RESPONSES (raw code blocks)
--- CODE BLOCK 1 ---

```
# Get Transfers: Javascript Example Request
async function getTransfers() {
 const response = await fetch("https://api.soccerdataapi.com/transfers/?
team_id=4138&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
--- END CODE BLOCK 1 ---

--- CODE BLOCK 2 ---
# Get Transfers: Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/transfers/?team_id=4138&auth_token=YOUR-AUTH-
TOKEN'
--- END CODE BLOCK 2 ---

--- CODE BLOCK 3 ---
# Get Transfers: Python Example Request
import requests
url = "https://api.soccerdataapi.com/transfers/"
querystring = {'team_id': 4138, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
--- END CODE BLOCK 3 ---

--- CODE BLOCK 4 ---
Get Transfers: Example JSON Response
{
 "id": 4138,
 "name": "Liverpool",
 "transfers": {
 "transfers_in": [
 {
 "player_id": 27537,
 "player_name": "L. Clarkson",
 "from_team": {
 "id": 2717,
 "name": "Aberdeen"
 },
 "transfer_date": "14-06-2023",
 "transfer_type": "n/a",
 "transfer_amount": 0,
 "transfer_currency": "usd"
 },
 {
 "player_id": 61790,
 "player_name": "A. Mac Allister",
 "from_team": {
 "id": 3200,
 "name": "Brighton & Hove Albion"
 },
 "transfer_date": "14-06-2023",
 "transfer_type": "transfer-fee",
 "transfer_amount": 42000000,
 "transfer_currency": "eur"
 },
 ...
 ],
 "transfers_out": [
 {
 "player_id": 27486,
 "player_name": "R. Williams",
 "to_team": {
 "id": 2717,
 "name": "Aberdeen"
 },
 "transfer_date": "28-06-2023",
 "transfer_type": "loan",
 "transfer_amount": 0,
 "transfer_currency": "usd"
 },
 {
 "player_id": 27537,
 "player_name": "L. Clarkson",
 "to_team": {
 "id": 2717,
 "name": "Aberdeen"
 },
 "transfer_date": "15-06-2023",
 "transfer_type": "n/a",
 "transfer_amount": 0,
 "transfer_currency": "usd"
 },
 ...
 ]
 }
}
```

--- END CODE BLOCK 4 ---

NARRATIVE TEXT (section plain text)

Get Transfers

# Get Transfers: Javascript Example Request

```
async function getTransfers() {
 const response = await fetch("https://api.soccerdataapi.com/transfers/?team_id=4138&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

# Get Transfers: Curl Example Request

```
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url 'https://api.soccerdataapi.com/transfers/?team_id=4138&auth_token=YOUR-AUTH-TOKEN'
```

# Get Transfers: Python Example Request

```
import requests
url = "https://api.soccerdataapi.com/transfers/"
querystring = {'team_id': 4138, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
```

```
print(response.json())
```

Retrieve transfers by team_id with a GET request to the endpoint:
https://api.soccerdataapi.com/transfers/

Get Transfers: Example JSON Response

```
{
 "id": 4138,
 "name": "Liverpool",
 "transfers": {
 "transfers_in": [
 {
 "player_id": 27537,
 "player_name": "L. Clarkson",
 "from_team": {
 "id": 2717,
 "name": "Aberdeen"
 },
 "transfer_date": "14-06-2023",
 "transfer_type": "n/a",
 "transfer_amount": 0,
 "transfer_currency": "usd"
 },
 {
 "player_id": 61790,
 "player_name": "A. Mac Allister",
 "from_team": {
 "id": 3200,
 "name": "Brighton & Hove Albion"
 },
 "transfer_date": "14-06-2023",
 "transfer_type": "transfer-fee",
 "transfer_amount": 42000000,
 "transfer_currency": "eur"
 },
 ...
 ],
 "transfers_out": [
 {
 "player_id": 27486,
 "player_name": "R. Williams",
 "to_team": {
 "id": 2717,
 "name": "Aberdeen"
 },
 "transfer_date": "28-06-2023",
 "transfer_type": "loan",
 "transfer_amount": 0,
 "transfer_currency": "usd"
 },
 {
 "player_id": 27537,
 "player_name": "L. Clarkson",
 "to_team": {
 "id": 2717,
 "name": "Aberdeen"
 },
 "transfer_date": "15-06-2023",
 "transfer_type": "n/a",
 "transfer_amount": 0,
 "transfer_currency": "usd"
 },
 ...
 ]
 }
}
```

```
QUERY PARAMETERS
Field
Type
Description
team_id
Integer
(required) Get transfers by team_id

================================================================================

[11] Get Head To Head
Anchor ID: content-get-head-to-head
--------------------------------------------------------------------------------
ENDPOINT URLS
- https://api.soccerdataapi.com/head-to-head/?
team_1_id=4137&team_2_id=4149&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/head-to-head/

QUERY PARAMETERS (table rows)
- team_1_id | Integer | (required) First team by team_id
- team_2_id | Integer | (required) Second team by team_id

CODE / EXAMPLES / RESPONSES (raw code blocks)
--- CODE BLOCK 1 ---
# Get Head To Head: Javascript Example Request
async function getHeadToHead() {
 const response = await fetch("https://api.soccerdataapi.com/head-to-head/?
team_1_id=4137&team_2_id=4149&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
--- END CODE BLOCK 1 ---

--- CODE BLOCK 2 ---
# Get Head To Head: Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/head-to-head/?
team_1_id=4137&team_2_id=4149&auth_token=YOUR-AUTH-TOKEN'
--- END CODE BLOCK 2 ---

--- CODE BLOCK 3 ---
# Get Head To Head: Python Example Request
import requests
url = "https://api.soccerdataapi.com/head-to-head/"
querystring = {'team_1_id': 4137, 'team_2_id': 4149, 'auth_token': YOUR-AUTH-
TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
```

```
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

--- END CODE BLOCK 3 ---

--- CODE BLOCK 4 ---

Get Head To Head: Example JSON Response

```
{
 "id": 2893,
 "team1": {
 "id": 4137,
 "name": "Manchester United"
 },
 "team2": {
 "id": 4149,
 "name": "Nottingham Forest"
 },
 "stats": {
 "overall": {
 "overall_games_played": 82,
 "overall_team1_wins": 41,
 "overall_team2_wins": 24,
 "overall_draws": 17,
 "overall_team1_scored": 153,
 "overall_team2_scored": 99
 },
 "team1_at_home": {
 "team1_games_played_at_home": 41,
 "team1_wins_at_home": 25,
 "team1_losses_at_home": 7,
 "team1_draws_at_home": 9,
 "team1_scored_at_home": 89,
 "team1_conceded_at_home": 42
 },
 "team2_at_home": {
 "team2_games_played_at_home": 41,
 "team2_wins_at_home": 17,
 "team2_losses_at_home": 16,
 "team2_draws_at_home": 8,
 "team2_scored_at_home": 57,
 "team2_conceded_at_home": 64
 }
 }
}
```

--- END CODE BLOCK 4 ---

NARRATIVE TEXT (section plain text)

Get Head To Head

```
# Get Head To Head: Javascript Example Request
async function getHeadToHead() {
 const response = await fetch("https://api.soccerdataapi.com/head-to-head/?
team_1_id=4137&team_2_id=4149&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

# Get Head To Head: Curl Example Request

```
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/head-to-head/?
team_1_id=4137&team_2_id=4149&auth_token=YOUR-AUTH-TOKEN'
```

# Get Head To Head: Python Example Request

```
import requests
url = "https://api.soccerdataapi.com/head-to-head/"
querystring = {'team_1_id': 4137, 'team_2_id': 4149, 'auth_token': YOUR-AUTH-
TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

Retrieve head to head stats by team_ids with a GET request to the endpoint:
https://api.soccerdataapi.com/head-to-head/

Get Head To Head: Example JSON Response

```
{
 "id": 2893,
 "team1": {
 "id": 4137,
 "name": "Manchester United"
 },
 "team2": {
 "id": 4149,
 "name": "Nottingham Forest"
 },
 "stats": {
 "overall": {
 "overall_games_played": 82,
 "overall_team1_wins": 41,
 "overall_team2_wins": 24,
 "overall_draws": 17,
 "overall_team1_scored": 153,
 "overall_team2_scored": 99
 },
 "team1_at_home": {
 "team1_games_played_at_home": 41,
 "team1_wins_at_home": 25,
 "team1_losses_at_home": 7,
 "team1_draws_at_home": 9,
 "team1_scored_at_home": 89,
 "team1_conceded_at_home": 42
 },
 "team2_at_home": {
 "team2_games_played_at_home": 41,
 "team2_wins_at_home": 17,
 "team2_losses_at_home": 16,
 "team2_draws_at_home": 8,
 "team2_scored_at_home": 57,
 "team2_conceded_at_home": 64
 }
 }
}
```

QUERY PARAMETERS

Field

Type

Description

team_1_id

```
Integer
(required) First team by team_id
team_2_id
Integer
(required) Second team by team_id

================================================================================

[12] Get Standing
Anchor ID: content-get-standing
--------------------------------------------------------------------------------
ENDPOINT URLS
- https://api.soccerdataapi.com/standing/?league_id=228&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/standing/

QUERY PARAMETERS (table rows)
- league_id | Integer | (required) Get standing by league_id
- season | String | (optional) Get standing by league season

CODE / EXAMPLES / RESPONSES (raw code blocks)
--- CODE BLOCK 1 ---
# Get Standing: Javascript Example Request
async function getStanding() {
 const response = await fetch("https://api.soccerdataapi.com/standing/?league_id=228&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
--- END CODE BLOCK 1 ---

--- CODE BLOCK 2 ---
# Get Standing Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/standing/?league_id=228&auth_token=YOUR-AUTH-TOKEN'
--- END CODE BLOCK 2 ---

--- CODE BLOCK 3 ---
# Get Standing: Python Example Request
import requests
url = "https://api.soccerdataapi.com/standing/"
querystring = {'league_id': 228, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
--- END CODE BLOCK 3 ---
```

--- CODE BLOCK 4 ---

Get Standing: Example JSON Response

```
{
 "id": 228,
 "league": {
 "id": 228,
 "name": "Premier League"
 },
 "season": "2023-2024",
 "stage": [
 {
 "stage_id": 6497,
 "stage_name": "Regular Season",
 "has_groups": false,
 "is_active": true,
 "standings": [
 {
 "position": 1,
 "team_id": 3059,
 "team_name": "West Ham United",
 "position_attribute": "Promotion - Champions League (Group Stage)",
 "games_played": 3,
 "points": 7,
 "wins": 2,
 "draws": 1,
 "losses": 0,
 "goals_for": 7,
 "goals_against": 3
 },
 {
 "position": 2,
 "team_id": 2909,
 "team_name": "Tottenham Hotspur",
 "position_attribute": "Promotion - Champions League (Group Stage)",
 "games_played": 3,
 "points": 7,
 "wins": 2,
 "draws": 1,
 "losses": 0,
 "goals_for": 6,
 "goals_against": 2
 },
 {
 "position": 3,
 "team_id": 3068,
 "team_name": "Arsenal",
 "position_attribute": "Promotion - Champions League (Group Stage)",
 "games_played": 3,
 "points": 7,
 "wins": 2,
 "draws": 1,
 "losses": 0,
 "goals_for": 5,
 "goals_against": 3
 },
 ...
 ]
 },
 ...
 ]
}
```

--- END CODE BLOCK 4 ---

NARRATIVE TEXT (section plain text)

Get Standing

# Get Standing: Javascript Example Request

```
async function getStanding() {
 const response = await fetch("https://api.soccerdataapi.com/standing/?league_id=228&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

# Get Standing Curl Example Request

```
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url 'https://api.soccerdataapi.com/standing/?league_id=228&auth_token=YOUR-AUTH-TOKEN'
```

# Get Standing: Python Example Request

```
import requests
url = "https://api.soccerdataapi.com/standing/"
querystring = {'league_id': 228, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

Retrieve standings by league id with a GET request to the endpoint:
https://api.soccerdataapi.com/standing/

Get Standing: Example JSON Response

```
{
 "id": 228,
 "league": {
 "id": 228,
 "name": "Premier League"
 },
 "season": "2023-2024",
 "stage": [
 {
 "stage_id": 6497,
 "stage_name": "Regular Season",
 "has_groups": false,
 "is_active": true,
 "standings": [
 {
 "position": 1,
 "team_id": 3059,
 "team_name": "West Ham United",
 "position_attribute": "Promotion - Champions League (Group Stage)",
 "games_played": 3,
 "points": 7,
 "wins": 2,
 "draws": 1,
 "losses": 0,
 "goals_for": 7,
 "goals_against": 3
 },
 {
 "position": 2,
 "team_id": 2909,
 "team_name": "Tottenham Hotspur",
 "position_attribute": "Promotion - Champions League (Group Stage)",
 "games_played": 3,
 "points": 7,
 "wins": 2,
 "draws": 1,
 "losses": 0,
 "goals_for": 6,
 "goals_against": 2
 },
 {
 "position": 3,
 "team_id": 3068,
 "team_name": "Arsenal",
 "position_attribute": "Promotion - Champions League (Group Stage)",
 "games_played": 3,
 "points": 7,
 "wins": 2,
 "draws": 1,
 "losses": 0,
 "goals_for": 5,
 "goals_against": 3
 },
 ...
 ]
 },
 ...
 ]
}
QUERY PARAMETERS
Field
Type
Description
league_id
Integer
(required) Get standing by league_id
season
String
(optional) Get standing by league season

================================================================================

[13] Get Live Scores
Anchor ID: content-get-livescores
--------------------------------------------------------------------------------
ENDPOINT URLS
- https://api.soccerdataapi.com/livescores/?auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/livescores/

CODE / EXAMPLES / RESPONSES (raw code blocks)
--- CODE BLOCK 1 ---
# Get Live Scores: Javascript Example Request
async function getLivescores() {
 const response = await fetch("https://api.soccerdataapi.com/livescores/?
auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
--- END CODE BLOCK 1 ---

--- CODE BLOCK 2 ---
# Get Live Scores Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/livescores/?auth_token=YOUR-AUTH-TOKEN'
--- END CODE BLOCK 2 ---

--- CODE BLOCK 3 ---
# Get Live Scores: Python Example Request
import requests
url = "https://api.soccerdataapi.com/livescores/"
querystring = {'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
--- END CODE BLOCK 3 ---

--- CODE BLOCK 4 ---
Get Live Scores: Example JSON Response
[
 {
 "league_id": 206,
 "league_name": "Liga Profesional",
 "country": {
 "id": 68,
 "name": "argentina"
 },
 "is_cup": false,
 "matches": [
 {
 "id": 531585,
 "stage_id": 6347,
 "date": "26/08/2023",
 "time": "00:30",
 "teams": {
 "home": {
 "id": 3842,
 "name": "Colon"
 },
 "away": {
 "id": 3843,
 "name": "Gimnasia La Plata"
 }
 },
 "stadium": {
 "id": 1891,
"name": "Estadio Brigadier General Estanislao Lopez",
"city": "Ciudad de Santa Fe, Provincia de Santa Fe"
},
"status": "finished",
"minute": -1,
"winner": "home",
"has_extra_time": false,
"has_penalties": false,
"goals": {
"home_ht_goals": 2,
"away_ht_goals": 0,
"home_ft_goals": 2,
"away_ft_goals": 0,
"home_et_goals": -1,
"away_et_goals": -1,
"home_pen_goals": -1,
"away_pen_goals": -1
},
"events": [
{
"event_type": "goal",
"event_minute": "14",
"team": "home",
"player": {
"id": 53675,
"name": "J. BenÃtez"
},
"assist_player": null
},
{
"event_type": "goal",
"event_minute": "27",
"team": "home",
"player": {
"id": 53644,
"name": "T. GalvÃ¡n"
},
"assist_player": null
},
{
"event_type": "yellow_card",
"event_minute": "30",
"team": "home",
"player": {
"id": 53590,
"name": "F. GarcÃ©s"
}
},
...
],
"odds": {
"match_winner": {
"home": 1.84,
"draw": 3.5,
"away": 4.3
},
"over_under": {
"total": 2.5,
"over": 2.1,
"under": 1.74
},
"handicap": {
"market": -0.5,
"home": 1.81,
"away": 1.96
},
"last_modified_timestamp": 1693017076
},
"lineups": {
"lineup_type": "live",
"lineups": {
"home": [
{
"player": {
"id": 102150,
"name": "R. Botta"
},
"position": "M"
},
{
"player": {
"id": 53656,
"name": "S. Moreyra"
},
"position": "M"
},
...
],
"away": [
{
"player": {
"id": 102150,
"name": "R. Botta"
},
"position": "M"
},
{
"player": {
"id": 53656,
"name": "S. Moreyra"
},
"position": "M"
},
...
]
},
"bench": {
"home": [
{
"player": {
"id": 53637,
"name": "B. Perlaza"
},
"position": "M"
},
{
"player": {
"id": 53653,
"name": "L. Picco"
},
"position": "M"
},
...
],
"away": [
{
"player": {
"id": 53692,
 "name": "Z. Zegarra"
 },
 "position": "M"
 },
 {
 "player": {
 "id": 84921,
 "name": "R. Saravia"
 },
 "position": "M"
 },
 ...
 ]
 },
 "sidelined": {
 "home": [
 {
 "player": {
 "id": 31889,
 "name": "M. Novak"
 },
 "status": "out",
 "desc": "Injury"
 }
 ],
 "away": [
 {
 "player": {
 "id": 31889,
 "name": "M. Novak"
 },
 "status": "out",
 "desc": "Injury"
 }
 ]
 },
 "formation": {
 "home": "4-3-3",
 "away": "4-3-3"
 }
 },
 "match_preview": {
 "has_preview": true,
 "word_count": 486
 }
 },
 ...
 ]
 },
 ...
]
```

--- END CODE BLOCK 4 ---

NARRATIVE TEXT (section plain text)

Get Live Scores

# Get Live Scores: Javascript Example Request

```
async function getLivescores() {
 const response = await fetch("https://api.soccerdataapi.com/livescores/?
auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

# Get Live Scores Curl Example Request

```
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/livescores/?auth_token=YOUR-AUTH-TOKEN'
```

# Get Live Scores: Python Example Request

```
import requests
url = "https://api.soccerdataapi.com/livescores/"
querystring = {'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

Retrieve live matches for current day (UTC) with a GET request to the endpoint:
https://api.soccerdataapi.com/livescores/

Get Live Scores: Example JSON Response

```
[
 {
 "league_id": 206,
 "league_name": "Liga Profesional",
 "country": {
 "id": 68,
 "name": "argentina"
 },
 "is_cup": false,
 "matches": [
 {
 "id": 531585,
 "stage_id": 6347,
 "date": "26/08/2023",
 "time": "00:30",
 "teams": {
 "home": {
 "id": 3842,
 "name": "Colon"
 },
 "away": {
 "id": 3843,
 "name": "Gimnasia La Plata"
 }
 },
 "stadium": {
 "id": 1891,
 "name": "Estadio Brigadier General Estanislao Lopez",
 "city": "Ciudad de Santa Fe, Provincia de Santa Fe"
 },
 "status": "finished",
 "minute": -1,
 "winner": "home",
 "has_extra_time": false,
 "has_penalties": false,
 "goals": {
```

```
"home_ht_goals": 2,
"away_ht_goals": 0,
"home_ft_goals": 2,
"away_ft_goals": 0,
"home_et_goals": -1,
"away_et_goals": -1,
"home_pen_goals": -1,
"away_pen_goals": -1
},
"events": [
{
"event_type": "goal",
"event_minute": "14",
"team": "home",
"player": {
"id": 53675,
"name": "J. BenÃtez"
},
"assist_player": null
},
{
"event_type": "goal",
"event_minute": "27",
"team": "home",
"player": {
"id": 53644,
"name": "T. GalvÃ¡n"
},
"assist_player": null
},
{
"event_type": "yellow_card",
"event_minute": "30",
"team": "home",
"player": {
"id": 53590,
"name": "F. GarcÃ©s"
}
},
...
],
"odds": {
"match_winner": {
"home": 1.84,
"draw": 3.5,
"away": 4.3
},
"over_under": {
"total": 2.5,
"over": 2.1,
"under": 1.74
},
"handicap": {
"market": -0.5,
"home": 1.81,
"away": 1.96
},
"last_modified_timestamp": 1693017076
},
"lineups": {
"lineup_type": "live",
"lineups": {
"home": [
{
```

```
"player": {
"id": 102150,
"name": "R. Botta"
},
"position": "M"
},
{
"player": {
"id": 53656,
"name": "S. Moreyra"
},
"position": "M"
},
...
],
"away": [
{
"player": {
"id": 102150,
"name": "R. Botta"
},
"position": "M"
},
{
"player": {
"id": 53656,
"name": "S. Moreyra"
},
"position": "M"
},
...
]
},
"bench": {
"home": [
{
"player": {
"id": 53637,
"name": "B. Perlaza"
},
"position": "M"
},
{
"player": {
"id": 53653,
"name": "L. Picco"
},
"position": "M"
},
...
],
"away": [
{
"player": {
"id": 53692,
"name": "Z. Zegarra"
},
"position": "M"
},
{
"player": {
"id": 84921,
"name": "R. Saravia"
},
 "position": "M"
 },
 ...
 ]
 },
 "sidelined": {
 "home": [
 {
 "player": {
 "id": 31889,
 "name": "M. Novak"
 },
 "status": "out",
 "desc": "Injury"
 }
 ],
 "away": [
 {
 "player": {
 "id": 31889,
 "name": "M. Novak"
 },
 "status": "out",
 "desc": "Injury"
 }
 ]
 },
 "formation": {
 "home": "4-3-3",
 "away": "4-3-3"
 }
 },
 "match_preview": {
 "has_preview": true,
 "word_count": 486
 }
 },
 ...
 ]
 },
 ...
]
```

==

[14] Get Matches

Anchor ID: content-get-matches

ENDPOINT URLS
- https://api.soccerdataapi.com/matches/?league_id=228&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/matches/

QUERY PARAMETERS (table rows)
- date | String | (optionally required) Get matches by date
- league_id | Integer | (optionally required) Get matches by league_id for current season
- league_id, season | String | (optional) Get matches by league and season
- league_id, date | String | (optional) Get matches by league and date

CODE / EXAMPLES / RESPONSES (raw code blocks)

--- CODE BLOCK 1 ---

```
# Get Matches: Javascript Example Request
async function getMatches() {
 const response = await fetch("https://api.soccerdataapi.com/matches/?league_id=228&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
--- END CODE BLOCK 1 ---
```

```
--- CODE BLOCK 2 ---
# Get Matches Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url 'https://api.soccerdataapi.com/matches/?league_id=228&auth_token=YOUR-AUTH-TOKEN'
--- END CODE BLOCK 2 ---
```

```
--- CODE BLOCK 3 ---
# Get Matches: Python Example Request
import requests
url = "https://api.soccerdataapi.com/matches/"
querystring = {'league_id': 228, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
--- END CODE BLOCK 3 ---
```

```
--- CODE BLOCK 4 ---
Get Matches: Example JSON Response
[
 {
 "league_id": 228,
 "league_name": "Premier League",
 "country": {
 "id": 8,
 "name": "england"
 },
 "is_cup": false,
 "matches": [
 {
 "id": 567518,
 "stage": {
 "id": 6497,
 "name": "Premier League"
 },
 "date": "11/08/2023",
 "time": "19:00",
 "teams": {
 "home": {
 "id": 3104,
 "name": "Burnley"
 },
 "away": {
 "id": 4136,
 "name": "Manchester City"
 }
 },
 "status": "finished",
 "minute": -1,
 "winner": "away",
 "has_extra_time": false,
 "has_penalties": false,
 "goals": {
 "home_ht_goals": 0,
 "away_ht_goals": 2,
 "home_ft_goals": 0,
 "away_ft_goals": 3,
 "home_et_goals": -1,
 "away_et_goals": -1,
 "home_pen_goals": -1,
 "away_pen_goals": -1
 },
 "odds": {
 "match_winner": {},
 "over_under": {},
 "handicap": {}
 },
 "match_preview": {
 "has_previews": false,
 "word_count": -1
 }
 },
 ...
 ]
 }
}
```

--- END CODE BLOCK 4 ---

NARRATIVE TEXT (section plain text)
Get Matches
# Get Matches: Javascript Example Request

```
async function getMatches() {
 const response = await fetch("https://api.soccerdataapi.com/matches/?
league_id=228&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

# Get Matches Curl Example Request

```
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/matches/?league_id=228&auth_token=YOUR-AUTH-
```

```
TOKEN'
# Get Matches: Python Example Request
import requests
url = "https://api.soccerdataapi.com/matches/"
querystring = {'league_id': 228, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

Retrieve matches by date or league id (optionally with season paramater) with a GET request to the endpoint:

https://api.soccerdataapi.com/matches/

Get Matches: Example JSON Response

```
[
 {
 "league_id": 228,
 "league_name": "Premier League",
 "country": {
 "id": 8,
 "name": "england"
 },
 "is_cup": false,
 "matches": [
 {
 "id": 567518,
 "stage": {
 "id": 6497,
 "name": "Premier League"
 },
 "date": "11/08/2023",
 "time": "19:00",
 "teams": {
 "home": {
 "id": 3104,
 "name": "Burnley"
 },
 "away": {
 "id": 4136,
 "name": "Manchester City"
 }
 },
 "status": "finished",
 "minute": -1,
 "winner": "away",
 "has_extra_time": false,
 "has_penalties": false,
 "goals": {
 "home_ht_goals": 0,
 "away_ht_goals": 2,
 "home_ft_goals": 0,
 "away_ft_goals": 3,
 "home_et_goals": -1,
 "away_et_goals": -1,
 "home_pen_goals": -1,
 "away_pen_goals": -1
 },
 "odds": {
 "match_winner": {},
 "over_under": {},
 "handicap": {}
 },
 "match_preview": {
```

```
 "has_previews": false,
 "word_count": -1
 }
 },
 ...
 ]
 }
}
```

QUERY PARAMETERS

Field	Type	Description
date	String	(optionally required) Get matches by date
league_id	Integer	(optionally required) Get matches by league_id for current season
league_id, season	String	(optional) Get matches by league and season
league_id, date	String	(optional) Get matches by league and date

==

[15] Get Match
Anchor ID: content-get-match

ENDPOINT URLS
- https://api.soccerdataapi.com/match/?match_id=531585&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/match/

QUERY PARAMETERS (table rows)
- match_id | Integer | (required) Get match by id

CODE / EXAMPLES / RESPONSES (raw code blocks)
--- CODE BLOCK 1 ---

```
# Get Match: Javascript Example Request
async function getMatch() {
 const response = await fetch("https://api.soccerdataapi.com/match/?
match_id=531585&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

--- END CODE BLOCK 1 ---

--- CODE BLOCK 2 ---

```
# Get Match Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/match/?match_id=531585&auth_token=YOUR-AUTH-
TOKEN'
--- END CODE BLOCK 2 ---

--- CODE BLOCK 3 ---
# Get Match: Python Example Request
import requests
url = "https://api.soccerdataapi.com/match/"
querystring = {'match_id': 531585, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
--- END CODE BLOCK 3 ---

--- CODE BLOCK 4 ---
Get Match: Example JSON Response
{
 "id": 531585,
 "league": {
 "id": 206,
 "name": "Liga Profesional"
 },
 "stage": {
 "id": 6347,
 "name": "Liga Profesional Argentina: 2nd Phase"
 },
 "date": "26/08/2023",
 "time": "00:30",
 "teams": {
 "home": {
 "id": 3842,
 "name": "Colon"
 },
 "away": {
 "id": 3843,
 "name": "Gimnasia La Plata"
 }
 },
 "stadium": {
 "id": 1891,
 "name": "Estadio Brigadier General Estanislao Lopez",
 "city": "Ciudad de Santa Fe, Provincia de Santa Fe"
 },
 "status": "finished",
 "minute": -1,
 "winner": "home",
 "has_extra_time": false,
 "has_penalties": false,
 "goals": {
 "home_ht_goals": 2,
 "away_ht_goals": 0,
 "home_ft_goals": 2,
 "away_ft_goals": 0,
 "home_et_goals": -1,
 "away_et_goals": -1,
 "home_pen_goals": -1,
 "away_pen_goals": -1
 },
 "events": [
{
"event_type": "goal",
"event_minute": "14",
"team": "home",
"player": {
"id": 53675,
"name": "J. BenÃtez"
},
"assist_player": null
},
{
"event_type": "goal",
"event_minute": "27",
"team": "home",
"player": {
"id": 53644,
"name": "T. GalvÃ¡n"
},
"assist_player": null
},
...
],
"odds": {
"match_winner": {
"home": 1.84,
"draw": 3.5,
"away": 4.3
},
"over_under": {
"total": 2.5,
"over": 2.1,
"under": 1.74
},
"handicap": {
"market": -0.5,
"home": 1.81,
"away": 1.96
},
"last_modified_timestamp": 1693017076
},
"lineups": {
"lineup_type": "live",
"lineups": {
"home": [
{
"player": {
"id": 102150,
"name": "R. Botta"
},
"position": "M"
},
{
"player": {
"id": 53656,
"name": "S. Moreyra"
},
"position": "M"
},
...
],
"away": [
{
"player": {
"id": 102150,
"name": "R. Botta"
},
"position": "M"
},
{
"player": {
"id": 53656,
"name": "S. Moreyra"
},
"position": "M"
},
...
]
},
"bench": {
"home": [
{
"player": {
"id": 53637,
"name": "B. Perlaza"
},
"position": "M"
},
{
"player": {
"id": 53653,
"name": "L. Picco"
},
"position": "M"
},
...
],
"away": [
{
"player": {
"id": 53692,
"name": "Z. Zegarra"
},
"position": "M"
},
{
"player": {
"id": 84921,
"name": "R. Saravia"
},
"position": "M"
},
...
]
},
"sidelined": {
"home": [
{
"player": {
"id": 31889,
"name": "M. Novak"
},
"status": "out",
"desc": "Injury"
}
],
"away": [
{
"player": {
 "id": 31889,
 "name": "M. Novak"
 },
 "status": "out",
 "desc": "Injury"
 }
 ]
 },
 "formation": {
 "home": "4-3-3",
 "away": "4-3-3"
 }
 },
 "match_preview": {
 "has_previews": true,
 "word_count": 389
 }
}
```

--- END CODE BLOCK 4 ---

NARRATIVE TEXT (section plain text)

Get Match

# Get Match: Javascript Example Request

```
async function getMatch() {
 const response = await fetch("https://api.soccerdataapi.com/match/?
match_id=531585&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

# Get Match Curl Example Request

```
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/match/?match_id=531585&auth_token=YOUR-AUTH-
TOKEN'
```

# Get Match: Python Example Request

```
import requests
url = "https://api.soccerdataapi.com/match/"
querystring = {'match_id': 531585, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

Retrieve match by id with a GET request to the endpoint:
https://api.soccerdataapi.com/match/

Get Match: Example JSON Response

```
{
 "id": 531585,
 "league": {
 "id": 206,
"name": "Liga Profesional"
},
"stage": {
"id": 6347,
"name": "Liga Profesional Argentina: 2nd Phase"
},
"date": "26/08/2023",
"time": "00:30",
"teams": {
"home": {
"id": 3842,
"name": "Colon"
},
"away": {
"id": 3843,
"name": "Gimnasia La Plata"
}
},
"stadium": {
"id": 1891,
"name": "Estadio Brigadier General Estanislao Lopez",
"city": "Ciudad de Santa Fe, Provincia de Santa Fe"
},
"status": "finished",
"minute": -1,
"winner": "home",
"has_extra_time": false,
"has_penalties": false,
"goals": {
"home_ht_goals": 2,
"away_ht_goals": 0,
"home_ft_goals": 2,
"away_ft_goals": 0,
"home_et_goals": -1,
"away_et_goals": -1,
"home_pen_goals": -1,
"away_pen_goals": -1
},
"events": [
{
"event_type": "goal",
"event_minute": "14",
"team": "home",
"player": {
"id": 53675,
"name": "J. BenÃtez"
},
"assist_player": null
},
{
"event_type": "goal",
"event_minute": "27",
"team": "home",
"player": {
"id": 53644,
"name": "T. GalvÃ¡n"
},
"assist_player": null
},
...
],
"odds": {
"match_winner": {
"home": 1.84,
"draw": 3.5,
"away": 4.3
},
"over_under": {
"total": 2.5,
"over": 2.1,
"under": 1.74
},
"handicap": {
"market": -0.5,
"home": 1.81,
"away": 1.96
},
"last_modified_timestamp": 1693017076
},
"lineups": {
"lineup_type": "live",
"lineups": {
"home": [
{
"player": {
"id": 102150,
"name": "R. Botta"
},
"position": "M"
},
{
"player": {
"id": 53656,
"name": "S. Moreyra"
},
"position": "M"
},
...
],
"away": [
{
"player": {
"id": 102150,
"name": "R. Botta"
},
"position": "M"
},
{
"player": {
"id": 53656,
"name": "S. Moreyra"
},
"position": "M"
},
...
]
},
"bench": {
"home": [
{
"player": {
"id": 53637,
"name": "B. Perlaza"
},
"position": "M"
},
{
"player": {
 "id": 53653,
 "name": "L. Picco"
 },
 "position": "M"
 },
 ...
 ],
 "away": [
 {
 "player": {
 "id": 53692,
 "name": "Z. Zegarra"
 },
 "position": "M"
 },
 {
 "player": {
 "id": 84921,
 "name": "R. Saravia"
 },
 "position": "M"
 },
 ...
 ]
 },
 "sidelined": {
 "home": [
 {
 "player": {
 "id": 31889,
 "name": "M. Novak"
 },
 "status": "out",
 "desc": "Injury"
 }
 ],
 "away": [
 {
 "player": {
 "id": 31889,
 "name": "M. Novak"
 },
 "status": "out",
 "desc": "Injury"
 }
 ]
 },
 "formation": {
 "home": "4-3-3",
 "away": "4-3-3"
 }
 },
 "match_preview": {
 "has_previews": true,
 "word_count": 389
 }
}
```

QUERY PARAMETERS

Field	Type	Description
match_id	Integer	(required) Get match by id

==

[16] Get Match Preview

Anchor ID: content-get-match-preview

ENDPOINT URLS

- https://api.soccerdataapi.com/match-preview/?match_id=544770&auth_token=YOUR-AUTH-TOKEN
- https://api.soccerdataapi.com/match-preview/

QUERY PARAMETERS (table rows)

- match_id | Integer | (required) Get preview by match_id

CODE / EXAMPLES / RESPONSES (raw code blocks)

--- CODE BLOCK 1 ---

```
# Get Match Preview: Javascript Example Request
async function getMatchPreview() {
 const response = await fetch("https://api.soccerdataapi.com/match-preview/?match_id=544770&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

--- END CODE BLOCK 1 ---

--- CODE BLOCK 2 ---

```
# Get Match Preview Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/match-preview/?match_id=544770&auth_token=YOUR-AUTH-TOKEN'
```

--- END CODE BLOCK 2 ---

--- CODE BLOCK 3 ---

```
# Get Match Preview: Python Example Request
import requests
url = "https://api.soccerdataapi.com/match-preview/"
querystring = {'match_id': 544770, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

--- END CODE BLOCK 3 ---

--- CODE BLOCK 4 ---

Get Match Preview: Example JSON Response

```
{
 "match_id": 544770,
 "league": {
```

```
 "id": 216,
 "name": "Serie B",
 "country": "brazil"
 },
 "home": {
 "id": 3958,
 "name": "Chapecoense"
 },
 "away": {
 "id": 3959,
 "name": "Avai"
 },
 "word_count": 362,
 "date": "27-08-2023",
 "time": "18:45",
 "match_data": {
 "weather": {
 "temp_f": 62.1,
 "temp_c": 16.7,
 "description": "sunny"
 },
 "excitement_rating": 5.53,
 "prediction": {
 "type": "match_winner",
 "choice": "Chapecoense Win"
 }
 },
 "content": [
 {
 "name": "p1",
 "content": "On Sunday, August 27, Chapecoense will face AvaÃ at Arena
CondÃ¡ Stadium in ChapecÃ³, Santa Catarina, at 18:45 (UTC) in the Brazil Serie B
league. This matchup marks a rematch of the teams' last game, a 1-4 win for
Chapecoense in the Serie B back on May 13. Fans in attendance can expect sunny
weather with a temperature of 62 degrees (F)."
 },
 {
 "name": "h1",
 "content": "Kayke's Rewards Reaped After Two-Goal Match"
 },
 {
 "name": "p2",
 "content": "Chapecoense have earned a total of 25 points in their last 24
matches, winning 6 and drawing 7 while losing 11. At home, they have won two and
drawn one of their last five matches, but have lacked quality in attack,
managing only 4 goals. Against similarly ranked opponents this year, they've had
a difficult run, winning none, drawing one and losing three, scoring an average
of 0.75 goals and conceding 1.75. In their last match, Kayke was instrumental in
a 1-2 away win against Botafogo SP, where he scored both goals."
 },
 {
 "name": "h2",
 "content": "Igor Bohn: The Reliable Road Goalkeeper with 10 Clean Sheets in 19
Games"
 },
 {
 "name": "p3",
 "content": "AvaÃ come into this game in good form, having won two of their last
five matches. In addition to their decent goal scoring record of 9 goals in
their last five outings, they will be entertained with a full strength squad.
Gabriel Poveda scored their lone goal when they drew 1-1 with CRB in the Serie B
last time out. Their imperious form on the road has been led by their
goalkeeper, Igor Bohn, who has kept ten clean sheets in 19 away matches this
season."
```

```
 },
 {
 "name": "p4",
 "content": "In the past 10 matches between Chapecoense and AvaÃ, an average of
2.3 goals have been scored. Out of 59 head-to-head meetings, Chapecoense has won
29, drawn 11 and AvaÃ has won 19 times."
 },
 {
 "name": "h3",
 "content": "Chapecoense Seeks to Leapfrog AvaÃ on League Table"
 },
 {
 "name": "p5",
 "content": "Chapecoense sit 1 point behind AvaÃ in the league table and have a
chance to jump them with their next game. This gives the home team plenty of
incentive to give it their all and strive to get the result they need. Going
into the match, Chapecoense will be highly motivated to score goals and solidify
their place at the top."
 }
 ]
}
```

--- END CODE BLOCK 4 ---

NARRATIVE TEXT (section plain text)

Get Match Preview

# Get Match Preview: Javascript Example Request

```
async function getMatchPreview() {
 const response = await fetch("https://api.soccerdataapi.com/match-preview/?
match_id=544770&auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

# Get Match Preview Curl Example Request

```
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/match-preview/?match_id=544770&auth_token=YOUR-
AUTH-TOKEN'
```

# Get Match Preview: Python Example Request

```
import requests
url = "https://api.soccerdataapi.com/match-preview/"
querystring = {'match_id': 544770, 'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

Retrieve match preview by match_id with a GET request to the endpoint:
https://api.soccerdataapi.com/match-preview/

Get Match Preview: Example JSON Response

```
{
 "match_id": 544770,
 "league": {
 "id": 216,
 "name": "Serie B",
 "country": "brazil"
 },
 "home": {
 "id": 3958,
 "name": "Chapecoense"
 },
 "away": {
 "id": 3959,
 "name": "Avai"
 },
 "word_count": 362,
 "date": "27-08-2023",
 "time": "18:45",
 "match_data": {
 "weather": {
 "temp_f": 62.1,
 "temp_c": 16.7,
 "description": "sunny"
 },
 "excitement_rating": 5.53,
 "prediction": {
 "type": "match_winner",
 "choice": "Chapecoense Win"
 }
 },
 "content": [
 {
 "name": "p1",
 "content": "On Sunday, August 27, Chapecoense will face AvaÃ at Arena
CondÃ¡ Stadium in ChapecÃ³, Santa Catarina, at 18:45 (UTC) in the Brazil Serie B
league. This matchup marks a rematch of the teams' last game, a 1-4 win for
Chapecoense in the Serie B back on May 13. Fans in attendance can expect sunny
weather with a temperature of 62 degrees (F)."
 },
 {
 "name": "h1",
 "content": "Kayke's Rewards Reaped After Two-Goal Match"
 },
 {
 "name": "p2",
 "content": "Chapecoense have earned a total of 25 points in their last 24
matches, winning 6 and drawing 7 while losing 11. At home, they have won two and
drawn one of their last five matches, but have lacked quality in attack,
managing only 4 goals. Against similarly ranked opponents this year, they've had
a difficult run, winning none, drawing one and losing three, scoring an average
of 0.75 goals and conceding 1.75. In their last match, Kayke was instrumental in
a 1-2 away win against Botafogo SP, where he scored both goals."
 },
 {
 "name": "h2",
 "content": "Igor Bohn: The Reliable Road Goalkeeper with 10 Clean Sheets in 19
Games"
 },
 {
 "name": "p3",
 "content": "AvaÃ come into this game in good form, having won two of their last
five matches. In addition to their decent goal scoring record of 9 goals in
their last five outings, they will be entertained with a full strength squad.
Gabriel Poveda scored their lone goal when they drew 1-1 with CRB in the Serie B
last time out. Their imperious form on the road has been led by their
goalkeeper, Igor Bohn, who has kept ten clean sheets in 19 away matches this
season."
 },
 {
 "name": "p4",
 "content": "In the past 10 matches between Chapecoense and AvaÃ, an average of
2.3 goals have been scored. Out of 59 head-to-head meetings, Chapecoense has won
29, drawn 11 and AvaÃ has won 19 times."
 },
 {
 "name": "h3",
 "content": "Chapecoense Seeks to Leapfrog AvaÃ on League Table"
 },
 {
 "name": "p5",
 "content": "Chapecoense sit 1 point behind AvaÃ in the league table and have a
chance to jump them with their next game. This gives the home team plenty of
incentive to give it their all and strive to get the result they need. Going
into the match, Chapecoense will be highly motivated to score goals and solidify
their place at the top."
 }
 ]
}
QUERY PARAMETERS
Field
Type
Description
match_id
Integer
(required) Get preview by match_id

================================================================================

[17] Get Upcoming Match Previews
Anchor ID: content-get-upcoming-match-previews
--------------------------------------------------------------------------------
ENDPOINT URLS
- https://api.soccerdataapi.com/match-previews-upcoming/?auth_token=YOUR-AUTH-
TOKEN
- https://api.soccerdataapi.com/match-previews-upcoming/

CODE / EXAMPLES / RESPONSES (raw code blocks)
--- CODE BLOCK 1 ---
# Get Upcoming Match Previews: Javascript Example Request
async function getUpcomingMatchPreviews() {
 const response = await fetch("https://api.soccerdataapi.com/match-previews-
upcoming/?auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
--- END CODE BLOCK 1 ---
```

--- CODE BLOCK 2 ---

```
# Get Upcoming Match Previews Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/match-previews-upcoming/?auth_token=YOUR-AUTH-TOKEN'
```

--- END CODE BLOCK 2 ---

--- CODE BLOCK 3 ---

```
# Get Upcoming Match Previews: Python Example Request
import requests
url = "https://api.soccerdataapi.com/match-previews-upcoming/"
querystring = {'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

--- END CODE BLOCK 3 ---

--- CODE BLOCK 4 ---

Get Upcoming Match Previews: Example JSON Response

```
[
 {
 "league_id": 216,
 "league_name": "Serie B",
 "country": {
 "id": 67,
 "name": "brazil"
 },
 "is_cup": false,
 "match_previews": [
 {
 "id": 544770,
 "date": "27/08/2023",
 "time": "18:45",
 "teams": {
 "home": {
 "id": 3958,
 "name": "Chapecoense"
 },
 "away": {
 "id": 3959,
 "name": "Avai"
 }
 },
 "word_count": 362,
 },
 ...
 ]
 },
 ...
]
```

--- END CODE BLOCK 4 ---

NARRATIVE TEXT (section plain text)

Get Upcoming Match Previews

```
# Get Upcoming Match Previews: Javascript Example Request
async function getUpcomingMatchPreviews() {
 const response = await fetch("https://api.soccerdataapi.com/match-previews-upcoming/?auth_token=YOUR-AUTH-TOKEN", {
 method: 'GET',
 headers: {
 "Content-Type": "application/json",
 "Accept-Encoding": "gzip"
 },
 })
 .then(response => {
 return response;
 })
 .catch(error => {
 return error;
 });
 const data = await response.json();
 console.log(data);
}
```

```
# Get Upcoming Match Previews Curl Example Request
curl --request GET \
 --compressed \
 --header 'Content-Type: application/json'--url
'https://api.soccerdataapi.com/match-previews-upcoming/?auth_token=YOUR-AUTH-TOKEN'
```

```
# Get Upcoming Match Previews: Python Example Request
import requests
url = "https://api.soccerdataapi.com/match-previews-upcoming/"
querystring = {'auth_token': YOUR-AUTH-TOKEN}
headers = {
 'Accept-Encoding': 'gzip',
 'Content-Type': 'application/json'
}
response = requests.get(url, headers=headers, params=querystring)
print(response.json())
```

Retrieve upcoming match previews with a GET request to the endpoint:
https://api.soccerdataapi.com/match-previews-upcoming/

Get Upcoming Match Previews: Example JSON Response

```
[
 {
 "league_id": 216,
 "league_name": "Serie B",
 "country": {
 "id": 67,
 "name": "brazil"
 },
 "is_cup": false,
 "match_previews": [
 {
 "id": 544770,
 "date": "27/08/2023",
 "time": "18:45",
 "teams": {
 "home": {
 "id": 3958,
 "name": "Chapecoense"
 },
 "away": {
 "id": 3959,
 "name": "Avai"
 }
 },
 "word_count": 362,
 },
 ...
 ]
 },
 ...
]
```

```
================================================================================

[18] Errors
Anchor ID: content-errors
--------------------------------------------------------------------------------
CODE / EXAMPLES / RESPONSES (raw code blocks)
--- CODE BLOCK 1 ---
# Invalid Request
--- END CODE BLOCK 1 ---

--- CODE BLOCK 2 ---
Error: Example JSON Response
{
 'detail': 'Error fetching match.'
}
--- END CODE BLOCK 2 ---

NARRATIVE TEXT (section plain text)
Errors
# Invalid Request
Invalid requests respond with a 200 status code, and an error message found in
the 'detail' attribute:
{"detail": "Invalid token."}
{"detail": "Request was throttled. Expected available in 60 seconds."}
{"detail": "Error fetching match.""}
Error: Example JSON Response
{
 'detail': 'Error fetching match.'
}

================================================================================
```