

DEVELOPER DOCUMENTATION

Sportsbook API Reference

Sports, live and pre-match events, leagues, tournaments and results through one simple, JSON-based HTTP API.

VERSION 2

REST + JSON

12 ENDPOINTS



```
GET https://w7api.com/sports/api/v2/inplay
    ?apikey=0
    &sid=1

200 OK application/json
```

API VERSION

v2

BASE URL

<https://w7api.com/sports/api/v2>

WEBSITE

bet365soft.com

Code examples included for [Node.js](#) [PHP](#) [Kotlin](#) [Java](#)

WELCOME

Overview

The **Sportsbook API** gives your application programmatic access to sports, live (in-play) events, pre-match events, leagues, tournaments, search and event results. Every endpoint is a simple HTTP **GET** request that returns **JSON**, so it works from any language or platform. This guide documents all 12 endpoints and includes ready-to-use client code for **Node.js**, **PHP**, **Kotlin** and **Java**.

API VERSION

v2

RESPONSE FORMAT

JSON

ENDPOINTS

12

AUTH

API key

Contents

Overview	2
Getting Started	3
Endpoint Overview	4
Endpoints	5
/sports - List sports	5
/inplay - Live events by sport	5
/inplayEvent - Live event details	6
/leagues - Leagues by sport	6
/prematch - Pre-match events	7
/prematchEvent - Pre-match event details	7
/upcoming - Upcoming events by date	8
/topMatch - Top matches	8
/search - Search events	9
/tournaments - List tournaments	9
/tournamentWinner - Tournament winner	9
/resultByEvent - Result by event	10
Code Examples	11
Node.js example	12
PHP example	13
Kotlin example	14
Java example	15
Best Practices & Notes	16

BASICS

Getting Started

Base URL

All requests are made to the following base URL. Append the endpoint path (for example `/sports`) and your query parameters.

```
BASE URL

https://w7api.com/sports/api/v2
```

Authentication

Every endpoint requires the `apikey` query parameter. If you do not have your own API key, pass `0` as the value.

Quick start

No registration is needed to try the API. Use `apikey=0`, copy the request below into a terminal, and you will receive the list of sports.

```
TERMINAL - cURL

curl "https://w7api.com/sports/api/v2/sports?apikey=0"
```

Request format

All endpoints use the **GET** method. Parameters are sent in the URL query string. Values with spaces or special characters must be URL-encoded (for example `Premier%20league`).

Parameter types

Type	Meaning
text	A string value, such as an API key, a date, a country code or a search keyword.
number	A numeric value, such as a sport ID, league ID or event ID.

Required and optional parameters

Each parameter in this guide is tagged with a **Required** or **Optional** badge. A request that omits a required parameter will not return valid data.

Typical workflows

Most integrations chain a few endpoints together. These are the common paths:

Browse pre-match events	<code>/sports</code>	,	<code>/leagues</code>	,	<code>/prematch</code>	,	<code>/prematchEvent</code>
Follow live events	<code>/sports</code>	,	<code>/inplay</code>	,	<code>/inplayEvent</code>		
Check a final result	Finished event ID	,	<code>/resultByEvent</code>				

REFERENCE

Endpoint Overview

A summary of all 12 endpoints. Click an endpoint in the Contents page for full details.

#	Endpoint	Description	Required parameters
1	/sports	List sports	apikey
2	/inplay	Live events by sport	apikey, sid
3	/inplayEvent	Live event details	apikey, id, oddType
4	/leagues	Leagues by sport	apikey, sid
5	/prematch	Pre-match events	apikey, lid, sid
6	/prematchEvent	Pre-match event details	apikey, id, oddType
7	/upcoming	Upcoming events by date	apikey, sid, fromDate, toDate
8	/topMatch	Top matches	apikey, status
9	/search	Search events	apikey, status, text
10	/tournaments	List tournaments	apikey
11	/tournamentWinner	Tournament winner	apikey, lid
12	/resultByEvent	Result by event	apikey, id

Tip: finding IDs

Sport IDs come from [/sports](#) , league IDs from [/leagues](#) (or [/tournaments](#)), and event IDs from the list endpoints such as [/inplay](#) , [/prematch](#) and [/upcoming](#) .

New in v2

The [/inplayEvent](#) and [/prematchEvent](#) endpoints now require the [oddType](#) parameter. Allowed values are [1](#) or [2](#) . All other endpoints work the same way as before.

REFERENCE

Endpoints

Each endpoint below lists its parameters and a ready-to-copy example request.

01

GET

/sports

List sports

Returns every sport supported by the API. Use the sport IDs from this response as the `sid` parameter in other endpoints.

Parameter	Type	Required	Default	Description
apikey	text	Required	0	Your API key. If you don't have your own key, enter 0 .



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v2/sports
?apikey=0
```

02

GET

/inplay

Live events by sport

Returns all in-play (live) events for the selected sport.

Parameter	Type	Required	Default	Description
apikey	text	Required	0	Your API key. If you don't have your own key, enter 0 .
sid	number	Required	1	Sport ID. Get valid IDs from /sports .



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v2/inplay
?apikey=0
&sid=1
```

03

GET

/inplayEvent

Live event details

Returns full details of a single in-play event. Choose the odds type with oddType.

Parameter	Type	Required	Default	Description
apikey	text	Required	0	Your API key. If you don't have your own key, enter 0.
id	number	Required	-	Event ID, for example one returned by /inplay .
oddType	number	Required	1	Odds type. Allowed values: 1 or 2 .

EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v2/inplayEvent
?apikey=0
&id={eventId}
&oddType=1
```

04

GET

/leagues

Leagues by sport

Returns the leagues available for the selected sport. League IDs are used as lid in pre-match requests.

Parameter	Type	Required	Default	Description
apikey	text	Required	0	Your API key. If you don't have your own key, enter 0.
sid	number	Required	1	Sport ID. Get valid IDs from /sports .

EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v2/leagues
?apikey=0
&sid=1
```

05

GET

/prematch

Pre-match events

Returns upcoming pre-match events for a league within a sport.

Parameter	Type	Required	Default	Description
apikey	text	Required	0	Your API key. If you don't have your own key, enter 0 .
lid	number	Required	-	League ID. Get valid IDs from /leagues .
sid	number	Required	-	Sport ID. Get valid IDs from /sports .

EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v2/prematch
?apikey=0
&lid={leagueId}
&sid=1
```

06

GET

/prematchEvent

Pre-match event details

Returns full details of a single pre-match event. Choose the odds type with oddType.

Parameter	Type	Required	Default	Description
apikey	text	Required	0	Your API key. If you don't have your own key, enter 0 .
id	number	Required	-	Event ID, for example one returned by /prematch .
oddType	number	Required	1	Odds type. Allowed values: 1 or 2 .

EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v2/prematchEvent
?apikey=0
&id={eventId}
&oddType=1
```

07

GET

/upcoming

Upcoming events by date

Returns events for a sport within a date range, with an optional country filter.

Parameter	Type	Required	Default	Description
apikey	text	Required	0	Your API key. If you don't have your own key, enter 0.
sid	number	Required	1	Sport ID. Get valid IDs from /sports .
fromDate	text	Required	today	Start of the date range, e.g. 2026-10-18.
toDate	text	Required	tomorrow	End of the date range, e.g. 2026-10-19.
country	text	Optional	NG	Country code to filter events, e.g. NG. Optional.



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v2/upcoming
?apikey=0
&sid=1
&fromDate=2026-10-18
&toDate=2026-10-19
&country=NG
```

08

GET

/topMatch

Top matches

Returns the featured top matches, either live or pre-match.

Parameter	Type	Required	Default	Description
apikey	text	Required	0	Your API key. If you don't have your own key, enter 0.
status	text	Required	Line	Event status. Allowed values: Live or Line .



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v2/topMatch
?apikey=0
&status=Line
```

09

GET

/search

Search events

Searches events by name, for example a league or a team.

Parameter	Type	Required	Default	Description
apikey	text	Required	0	Your API key. If you don't have your own key, enter 0.
status	text	Required	Line	Event status. Allowed values: <code>Live</code> or <code>Line</code> .
text	text	Required	Premier league	Search keyword. URL-encode spaces (e.g. Premier%20league).



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v2/search
?apikey=0
&status=Line
&text=Premier%20league
```

10

GET

/tournaments

List tournaments

Returns the list of tournaments available through the API.

Parameter	Type	Required	Default	Description
apikey	text	Required	0	Your API key. If you don't have your own key, enter 0.



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v2/tournaments
?apikey=0
```

11

GET

/tournamentWinner

Tournament winner

Returns the tournament-winner (outright) data for a championship.

Parameter	Type	Required	Default	Description
apikey	text	Required	0	Your API key. If you don't have your own key, enter 0.
lid	number	Required	1	Championship (league) ID.



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v2/tournamentWinner
?apikey=0
&lid={leagueId}
```

12

GET

/resultByEvent

Result by event

Returns the result of a finished event.

Parameter	Type	Required	Default	Description
apikey	text	Required	0	Your API key. If you don't have your own key, enter 0.
id	number	Required	1	ID of the finished event.

EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v2/resultByEvent
?apikey=0
&id={eventId}
```

INTEGRATION

Code Examples

Each example below is a complete, self-contained client with one method per endpoint, plus a short usage section at the bottom. Copy it into your project, set your API key (or keep `0`) and run.

Node.js

Node.js 18 or newer. No packages to install.

PHP

PHP 7.4 or newer with the cURL extension enabled.

Kotlin

Kotlin with the OkHttp library (Gradle dependency shown in the file).

Java

Java 11 or newer. Uses the built-in `java.net.http` client.

Keep your API key private

If you use your own API key, do not hardcode it in a public repository or ship it inside a mobile app. Load it from an environment variable or call the API from your backend.

CODE EXAMPLE

Node.js

 Node.js 18 or newer. No packages to install.Run: `node sportsbook-client.js`

```
NODE.JS - SportsbookClient

1 // sportsbook-client.js - Node.js 18+ (uses the built-in fetch)
2 const BASE_URL = "https://w7api.com/sports/api/v2";
3
4 class SportsbookClient {
5   constructor(apiKey = "0") {           // "0" = no personal key
6     this.apiKey = apiKey;
7   }
8
9   async #get(endpoint, params = {}) {
10    const query = new URLSearchParams({ apiKey: this.apiKey, ...params });
11    const res = await fetch(`${BASE_URL}/${endpoint}?${query}`);
12    if (!res.ok) throw new Error(`HTTP ${res.status} on /${endpoint}`);
13    return res.json();
14  }
15
16  sports() { return this.#get("sports"); }
17  inplay(sid = 1) { return this.#get("inplay", { sid }); }
18  inplayEvent(id, oddType = 1) { return this.#get("inplayEvent", { id, oddType }); }
19  leagues(sid = 1) { return this.#get("leagues", { sid }); }
20  prematch(lid, sid) { return this.#get("prematch", { lid, sid }); }
21  prematchEvent(id, oddType = 1) { return this.#get("prematchEvent", { id, oddType }); }
22  upcoming(sid, fromDate, toDate, country) {
23    const p = { sid, fromDate, toDate };
24    if (country) p.country = country;           // optional
25    return this.#get("upcoming", p);
26  }
27  topMatch(status = "Line") { return this.#get("topMatch", { status }); }
28  search(text, status = "Line") { return this.#get("search", { status, text }); }
29  tournaments() { return this.#get("tournaments"); }
30  tournamentWinner(lid) { return this.#get("tournamentWinner", { lid }); }
31  resultByEvent(id) { return this.#get("resultByEvent", { id }); }
32 }
33
34 // ----- usage -----
35 (async () => {
36   const api = new SportsbookClient("0");
37   console.log(await api.sports());
38   console.log(await api.inplay(1));
39   console.log(await api.upcoming(1, "2026-10-18", "2026-10-19", "NG"));
40   console.log(await api.search("Premier league", "Line"));
41 })().catch(console.error);
```

CODE EXAMPLE

PHP

PHP 7.4 or newer with the cURL extension enabled.

Run: `php SportsbookClient.php`

```

1  <?php
2  // SportsbookClient.php - PHP 7.4+ (requires the cURL extension)
3
4  class SportsbookClient
5  {
6      private const BASE_URL = 'https://w7api.com/sports/api/v2';
7      private string $apiKey;
8
9      public function __construct(string $apiKey = '0')    // '0' = no personal key
10     {
11         $this->apiKey = $apiKey;
12     }
13
14     private function get(string $endpoint, array $params = []): array
15     {
16         $params = ['apikey' => $this->apiKey] + $params;
17         $params = array_filter($params, fn($v) => $v !== null && $v !== '');
18         $url = self::BASE_URL . '/' . $endpoint . '?' . http_build_query($params);
19
20         $ch = curl_init($url);
21         curl_setopt_array($ch, [CURLOPT_RETURNTRANSFER => true, CURLOPT_TIMEOUT => 15]);
22         $body = curl_exec($ch);
23         $code = curl_getinfo($ch, CURLINFO_HTTP_CODE);
24         $err = curl_error($ch);
25         curl_close($ch);
26
27         if ($body === false) throw new RuntimeException("cURL error: $err");
28         if ($code !== 200) throw new RuntimeException("HTTP $code on /$endpoint");
29         return json_decode($body, true, 512, JSON_THROW_ON_ERROR);
30     }
31
32     public function sports(): array { return $this->get('sports'); }
33     public function inplay(int $sid = 1): array { return $this->get('inplay', ['sid' => $sid]); }
34     public function inplayEvent(int $id, int $oddType = 1): array {
35         return $this->get('inplayEvent', ['id' => $id, 'oddType' => $oddType]);
36     }
37     public function leagues(int $sid = 1): array { return $this->get('leagues', ['sid' => $sid]); }
38     public function prematch(int $lid, int $sid): array {
39         return $this->get('prematch', ['lid' => $lid, 'sid' => $sid]);
40     }
41     public function prematchEvent(int $id, int $oddType = 1): array {
42         return $this->get('prematchEvent', ['id' => $id, 'oddType' => $oddType]);
43     }
44     public function upcoming(int $sid, string $from, string $to, ?string $country = null): array {
45         return $this->get('upcoming', ['sid' => $sid, 'fromDate' => $from,
46             'toDate' => $to, 'country' => $country]);
47     }
48     public function topMatch(string $status = 'Line'): array {
49         return $this->get('topMatch', ['status' => $status]);
50     }
51     public function search(string $text, string $status = 'Line'): array {
52         return $this->get('search', ['status' => $status, 'text' => $text]);
53     }
54     public function tournaments(): array { return $this->get('tournaments'); }
55     public function tournamentWinner(int $lid): array {
56         return $this->get('tournamentWinner', ['lid' => $lid]);
57     }
58     public function resultByEvent(int $id): array {
59         return $this->get('resultByEvent', ['id' => $id]);
60     }
61 }
62
63 // ----- usage -----
64 $api = new SportsbookClient('0');
65 print_r($api->sports());
66 print_r($api->inplay(1));
67 print_r($api->upcoming(1, '2026-10-18', '2026-10-19', 'NG'));
68 print_r($api->search('Premier league', 'Line'));

```

CODE EXAMPLE

Kotlin

 Kotlin with the OkHttp library (Gradle dependency shown in the file).**Run:** Run main() from your IDE or Gradle.

```

1 // SportsbookClient.kt - Kotlin + OkHttp 4.x
2 // Gradle: implementation("com.squareup.okhttp3:okhttp:4.12.0")
3 // Android: call these methods from a background thread / coroutine (Dispatchers.IO).
4 import okhttp3.HttpUrl.Companion.toHttpUrl
5 import okhttp3.OkHttpClient
6 import okhttp3.Request
7 import java.io.IOException
8
9 class SportsbookClient(private val apiKey: String = "0") { // "0" = no personal key
10
11     private val http = OkHttpClient()
12     private val baseUrl = "https://w7api.com/sports/api/v2"
13
14     /** Returns the raw JSON string. Parse it with Gson, Moshi or kotlinx.serialization. */
15     private fun get(endpoint: String, params: Map<String, Any?> = emptyMap()): String {
16         val url = "$baseUrl/$endpoint".toHttpUrl().newBuilder().apply {
17             addQueryParameter("apikey", apiKey)
18             params.forEach { (k, v) -> if (v != null) addQueryParameter(k, v.toString()) }
19         }.build()
20
21         http.newCall(Request.Builder().url(url).build()).execute().use { res ->
22             if (!res.isSuccessful) throw IOException("HTTP ${res.code} on /$endpoint")
23             return res.body!!.string()
24         }
25     }
26
27     fun sports() = get("sports")
28     fun inplay(sid: Int = 1) = get("inplay", mapOf("sid" to sid))
29     fun inplayEvent(id: Long, oddType: Int = 1) =
30         get("inplayEvent", mapOf("id" to id, "oddType" to oddType))
31     fun leagues(sid: Int = 1) = get("leagues", mapOf("sid" to sid))
32     fun prematch(lid: Long, sid: Int) = get("prematch", mapOf("lid" to lid, "sid" to sid))
33     fun prematchEvent(id: Long, oddType: Int = 1) =
34         get("prematchEvent", mapOf("id" to id, "oddType" to oddType))
35     fun upcoming(sid: Int, fromDate: String, toDate: String, country: String? = null) =
36         get("upcoming", mapOf("sid" to sid, "fromDate" to fromDate,
37             "toDate" to toDate, "country" to country))
38     fun topMatch(status: String = "Line") = get("topMatch", mapOf("status" to status))
39     fun search(text: String, status: String = "Line") =
40         get("search", mapOf("status" to status, "text" to text))
41     fun tournaments() = get("tournaments")
42     fun tournamentWinner(lid: Long) = get("tournamentWinner", mapOf("lid" to lid))
43     fun resultByEvent(id: Long) = get("resultByEvent", mapOf("id" to id))
44 }
45
46 // ----- usage -----
47 fun main() {
48     val api = SportsbookClient()
49     println(api.sports())
50     println(api.inplay(1))
51     println(api.upcoming(1, "2026-10-18", "2026-10-19", "NG"))
52     println(api.search("Premier league", "Line"))
53 }

```

CODE EXAMPLE

Java

 Java 11 or newer. Uses the built-in java.net.http client.Run: `javac SportsbookClient.java && java SportsbookClient`

JAVA - SportsbookClient

```

1  // SportsbookClient.java - Java 11+ (no external dependencies)
2  import java.io.IOException;
3  import java.net.URI;
4  import java.net.URLEncoder;
5  import java.net.http.*;
6  import java.nio.charset.StandardCharsets;
7  import java.time.Duration;
8  import java.util.*;
9  public class SportsbookClient {
10     private static final String BASE_URL = "https://w7api.com/sports/api/v2";
11     private final HttpClient http = HttpClient.newBuilder()
12         .connectTimeout(Duration.ofSeconds(10)).build();
13     private final String apiKey;
14     public SportsbookClient() { this(""); } // "" = no personal key
15     public SportsbookClient(String apiKey) { this.apiKey = apiKey; }
16
17     private static String enc(String s) { return URLEncoder.encode(s, StandardCharsets.UTF_8); }
18     /** Returns the raw JSON string. Parse it with Jackson, Gson or org.json. */
19     private String get(String endpoint, Map<String, Object> params) {
20         StringJoiner query = new StringJoiner("&");
21         query.add("apikey=" + enc(apiKey));
22         params.forEach((k, v) -> { if (v != null) query.add(k + "=" + enc(String.valueOf(v))); });
23         try {
24             HttpRequest req = HttpRequest
25                 .newBuilder(URI.create(BASE_URL + "/" + endpoint + "?" + query))
26                 .timeout(Duration.ofSeconds(15)).GET().build();
27             HttpResponse<String> res = http.send(req, HttpResponse.BodyHandlers.ofString());
28             if (res.statusCode() != 200) throw new IOException("HTTP " + res.statusCode());
29             return res.body();
30         } catch (IOException | InterruptedException e) {
31             throw new RuntimeException("Request failed: /" + endpoint, e);
32         }
33     }
34     public String sports() { return get("sports", Map.of()); }
35     public String inplay(int sid) { return get("inplay", Map.of("sid", sid)); }
36     public String inplayEvent(long id, int oddType) {
37         return get("inplayEvent", Map.of("id", id, "oddType", oddType));
38     }
39     public String leagues(int sid) { return get("leagues", Map.of("sid", sid)); }
40     public String prematch(long lid, int sid) {
41         return get("prematch", Map.of("lid", lid, "sid", sid));
42     }
43     public String prematchEvent(long id, int oddType) {
44         return get("prematchEvent", Map.of("id", id, "oddType", oddType));
45     }
46     public String upcoming(int sid, String from, String to, String country) {
47         Map<String, Object> p = new LinkedHashMap<>();
48         p.put("sid", sid); p.put("fromDate", from); p.put("toDate", to);
49         if (country != null) p.put("country", country); // optional
50         return get("upcoming", p);
51     }
52     public String topMatch(String status) { return get("topMatch", Map.of("status", status)); }
53     public String search(String text, String status) {
54         return get("search", Map.of("status", status, "text", text));
55     }
56     public String tournaments() { return get("tournaments", Map.of()); }
57     public String tournamentWinner(long lid) { return get("tournamentWinner", Map.of("lid", lid)); }
58     public String resultByEvent(long id) { return get("resultByEvent", Map.of("id", id)); }
59
60     public static void main(String[] args) { // usage
61         SportsbookClient api = new SportsbookClient();
62         System.out.println(api.sports());
63         System.out.println(api.inplay(1));
64         System.out.println(api.upcoming(1, "2026-10-18", "2026-10-19", "NG"));
65         System.out.println(api.search("Premier league", "Line"));
66     }
67 }

```

GUIDANCE

Best Practices & Notes

- 1 Cache slow-changing lists**
Responses from `/sports`, `/leagues` and `/tournaments` change rarely. Cache them for a while instead of requesting them on every page load.
- 2 Poll live data sensibly**
For `/inplay` and `/inplayEvent` refresh at a reasonable interval, and stop polling when the user leaves the screen.
- 3 Always URL-encode values**
Use your language's query builder (as in the examples) so values such as `Premier league` are encoded correctly.
- 4 Handle errors and timeouts**
Check the HTTP status code, set a request timeout and show a friendly message (or retry once) when a request fails.
- 5 Validate required parameters**
Make sure every required parameter is present before sending the request, especially IDs such as `sid`, `lid` and `id`.
- 6 Use the right status value**
`/topMatch` and `/search` accept only `Live` or `Line` as the `status` parameter.

HTTP status handling

Status	Meaning	What to do
200	Request succeeded.	Parse the JSON body.
4xx	The request is invalid, for example a missing or wrong parameter.	Check the parameters and the endpoint path.
5xx	A temporary problem on the server side.	Wait a moment and retry.

Ready to build?

Start with `/sports` using `apikey=0`. For more information visit bet365soft.com.