BET365SOFT.COM

DEVELOPER DOCUMENTATION

Sportsbook API Reference

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

VERSION 1 | REST + JSON | 12 ENDPOINTS

```
GET https://w7api.com/sports/api/v1/inplay
   ?apikey=0
   &sid=1

200 OK  application/json
```

API VERSION	BASE URL	WEBSITE
v1	https://w7api.com/sports/api/v1	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	RESPONSE FORMAT	ENDPOINTS	AUTH
v1	JSON	12	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/v1
```

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/v1/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	`/sports`	›	`/leagues`	›	`/prematch`	›	`/prematchEvent`
Follow live events	`/sports`	›	`/inplay`	›	`/inplayEvent`		
Check a final result	`Finished event ID`	›	`/resultByEvent`				

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`
4	`/leagues`	Leagues by sport	`apikey, sid`
5	`/prematch`	Pre-match events	`apikey, lid, sid`
6	`/prematchEvent`	Pre-match event details	`apikey, id`
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`.

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/v1/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/v1/inplay
    ?apikey=0
    &sid=1
```

03 GET `/inplayEvent`

Live event details

Returns full details of a single in-play 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**	-	Event ID, for example one returned by `/inplay`.

EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v1/inplayEvent
    ?apikey=0
    &id={eventId}
```

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/v1/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/v1/prematch
    ?apikey=0
    &lid={leagueId}
    &sid=1
```

06 GET `/prematchEvent` Pre-match event details

Returns full details of a single pre-match 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**	-	Event ID, for example one returned by `/prematch`.

EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v1/prematchEvent
    ?apikey=0
    &id={eventId}
```

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/v1/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/v1/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: `Live` or `Line`.
text	text	**Required**	`Premier league`	Search keyword. URL-encode spaces (e.g. Premier%20league).

EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v1/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/v1/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/v1/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/v1/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	PHP	Kotlin	Java
Node.js 18 or newer. No packages to install.	PHP 7.4 or newer with the cURL extension enabled.	Kotlin with the OkHttp library (Gradle dependency shown in the file).	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

```
// sportsbook-client.js  -  Node.js 18+ (uses the built-in fetch)
const BASE_URL = "https://w7api.com/sports/api/v1";

class SportsbookClient {
  constructor(apiKey = "0") {          // "0" = no personal key
    this.apiKey = apiKey;
  }

  async #get(endpoint, params = {}) {
    const query = new URLSearchParams({ apikey: this.apiKey, ...params });
    const res = await fetch(`${BASE_URL}/${endpoint}?${query}`);
    if (!res.ok) throw new Error(`HTTP ${res.status} on /${endpoint}`);
    return res.json();
  }

  sports()                      { return this.#get("sports"); }
  inplay(sid = 1)               { return this.#get("inplay", { sid }); }
  inplayEvent(id)               { return this.#get("inplayEvent", { id }); }
  leagues(sid = 1)              { return this.#get("leagues", { sid }); }
  prematch(lid, sid)            { return this.#get("prematch", { lid, sid }); }
  prematchEvent(id)             { return this.#get("prematchEvent", { id }); }
  upcoming(sid, fromDate, toDate, country) {
    const p = { sid, fromDate, toDate };
    if (country) p.country = country;            // optional
    return this.#get("upcoming", p);
  }
  topMatch(status = "Line")     { return this.#get("topMatch", { status }); }
  search(text, status = "Line") { return this.#get("search", { status, text }); }
  tournaments()                 { return this.#get("tournaments"); }
  tournamentWinner(lid)         { return this.#get("tournamentWinner", { lid }); }
  resultByEvent(id)             { return this.#get("resultByEvent", { id }); }
}

// ---------------------------- usage ----------------------------
(async () => {
  const api = new SportsbookClient("0");
  console.log(await api.sports());
  console.log(await api.inplay(1));
  console.log(await api.upcoming(1, "2026-10-18", "2026-10-19", "NG"));
  console.log(await api.search("Premier league", "Line"));
})().catch(console.error);
```

CODE EXAMPLE

PHP

PHP 7.4 or newer with the cURL extension enabled.

Run: `php SportsbookClient.php`

PHP - SportsbookClient

```php
<?php
// SportsbookClient.php  -  PHP 7.4+ (requires the cURL extension)

class SportsbookClient
{
    private const BASE_URL = 'https://w7api.com/sports/api/v1';
    private string $apiKey;

    public function __construct(string $apiKey = '0')   // '0' = no personal key
    {
        $this->apiKey = $apiKey;
    }

    private function get(string $endpoint, array $params = []): array
    {
        $params = ['apikey' => $this->apiKey] + $params;
        $params = array_filter($params, fn($v) => $v !== null && $v !== '');
        $url = self::BASE_URL . '/' . $endpoint . '?' . http_build_query($params);

        $ch = curl_init($url);
        curl_setopt_array($ch, [CURLOPT_RETURNTRANSFER => true, CURLOPT_TIMEOUT => 15]);
        $body = curl_exec($ch);
        $code = curl_getinfo($ch, CURLINFO_HTTP_CODE);
        $err  = curl_error($ch);
        curl_close($ch);

        if ($body === false) throw new RuntimeException("cURL error: $err");
        if ($code !== 200)   throw new RuntimeException("HTTP $code on /$endpoint");
        return json_decode($body, true, 512, JSON_THROW_ON_ERROR);
    }

    public function sports(): array { return $this->get('sports'); }
    public function inplay(int $sid = 1): array { return $this->get('inplay', ['sid' => $sid]); }
    public function inplayEvent(int $id): array {
        return $this->get('inplayEvent', ['id' => $id]);
    }
    public function leagues(int $sid = 1): array { return $this->get('leagues', ['sid' => $sid]); }
    public function prematch(int $lid, int $sid): array {
        return $this->get('prematch', ['lid' => $lid, 'sid' => $sid]);
    }
    public function prematchEvent(int $id): array {
        return $this->get('prematchEvent', ['id' => $id]);
    }
    public function upcoming(int $sid, string $from, string $to, ?string $country = null): array {
        return $this->get('upcoming', ['sid' => $sid, 'fromDate' => $from,
                                       'toDate' => $to, 'country' => $country]);
    }
    public function topMatch(string $status = 'Line'): array {
        return $this->get('topMatch', ['status' => $status]);
    }
    public function search(string $text, string $status = 'Line'): array {
        return $this->get('search', ['status' => $status, 'text' => $text]);
    }
    public function tournaments(): array { return $this->get('tournaments'); }
    public function tournamentWinner(int $lid): array {
        return $this->get('tournamentWinner', ['lid' => $lid]);
    }
    public function resultByEvent(int $id): array {
        return $this->get('resultByEvent', ['id' => $id]);
    }
}

// ----------------------------- usage -----------------------------
$api = new SportsbookClient('0');
print_r($api->sports());
print_r($api->inplay(1));
print_r($api->upcoming(1, '2026-10-18', '2026-10-19', 'NG'));
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.`

KOTLIN - SportsbookClient

```kotlin
// SportsbookClient.kt  -  Kotlin + OkHttp 4.x
// Gradle: implementation("com.squareup.okhttp3:okhttp:4.12.0")
// Android: call these methods from a background thread / coroutine (Dispatchers.IO).
import okhttp3.HttpUrl.Companion.toHttpUrl
import okhttp3.OkHttpClient
import okhttp3.Request
import java.io.IOException

class SportsbookClient(private val apiKey: String = "0") {   // "0" = no personal key

    private val http = OkHttpClient()
    private val baseUrl = "https://w7api.com/sports/api/v1"

    /** Returns the raw JSON string. Parse it with Gson, Moshi or kotlinx.serialization. */
    private fun get(endpoint: String, params: Map<String, Any?> = emptyMap()): String {
        val url = "$baseUrl/$endpoint".toHttpUrl().newBuilder().apply {
            addQueryParameter("apikey", apiKey)
            params.forEach { (k, v) -> if (v != null) addQueryParameter(k, v.toString()) }
        }.build()

        http.newCall(Request.Builder().url(url).build()).execute().use { res ->
            if (!res.isSuccessful) throw IOException("HTTP ${res.code} on /$endpoint")
            return res.body!!.string()
        }
    }

    fun sports() = get("sports")
    fun inplay(sid: Int = 1) = get("inplay", mapOf("sid" to sid))
    fun inplayEvent(id: Long) = get("inplayEvent", mapOf("id" to id))
    fun leagues(sid: Int = 1) = get("leagues", mapOf("sid" to sid))
    fun prematch(lid: Long, sid: Int) = get("prematch", mapOf("lid" to lid, "sid" to sid))
    fun prematchEvent(id: Long) = get("prematchEvent", mapOf("id" to id))
    fun upcoming(sid: Int, fromDate: String, toDate: String, country: String? = null) =
        get("upcoming", mapOf("sid" to sid, "fromDate" to fromDate,
                              "toDate" to toDate, "country" to country))
    fun topMatch(status: String = "Line") = get("topMatch", mapOf("status" to status))
    fun search(text: String, status: String = "Line") =
        get("search", mapOf("status" to status, "text" to text))
    fun tournaments() = get("tournaments")
    fun tournamentWinner(lid: Long) = get("tournamentWinner", mapOf("lid" to lid))
    fun resultByEvent(id: Long) = get("resultByEvent", mapOf("id" to id))
}

// ---------------------------- usage ----------------------------
fun main() {
    val api = SportsbookClient()
    println(api.sports())
    println(api.inplay(1))
    println(api.upcoming(1, "2026-10-18", "2026-10-19", "NG"))
    println(api.search("Premier league", "Line"))
}
```

CODE EXAMPLE

Java

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

Run: `javac SportsbookClient.java && java SportsbookClient`

JAVA - SportsbookClient

```java
// SportsbookClient.java  -  Java 11+ (no external dependencies)
import java.io.IOException;
import java.net.URI;
import java.net.URLEncoder;
import java.net.http.*;
import java.nio.charset.StandardCharsets;
import java.time.Duration;
import java.util.*;

public class SportsbookClient {
    private static final String BASE_URL = "https://w7api.com/sports/api/v1";
    private final HttpClient http = HttpClient.newBuilder()
            .connectTimeout(Duration.ofSeconds(10)).build();
    private final String apiKey;

    public SportsbookClient() { this("0"); }                  // "0" = no personal key
    public SportsbookClient(String apiKey) { this.apiKey = apiKey; }

    private static String enc(String s) { return URLEncoder.encode(s, StandardCharsets.UTF_8); }

    /** Returns the raw JSON string. Parse it with Jackson, Gson or org.json. */
    private String get(String endpoint, Map<String, Object> params) {
        StringJoiner query = new StringJoiner("&");
        query.add("apikey=" + enc(apiKey));
        params.forEach((k, v) -> { if (v != null) query.add(k + "=" + enc(String.valueOf(v))); });
        try {
            HttpRequest req = HttpRequest
                    .newBuilder(URI.create(BASE_URL + "/" + endpoint + "?" + query))
                    .timeout(Duration.ofSeconds(15)).GET().build();
            HttpResponse<String> res = http.send(req, HttpResponse.BodyHandlers.ofString());
            if (res.statusCode() != 200) throw new IOException("HTTP " + res.statusCode());
            return res.body();
        } catch (IOException | InterruptedException e) {
            throw new RuntimeException("Request failed: /" + endpoint, e);
        }
    }

    public String sports()                  { return get("sports", Map.of()); }
    public String inplay(int sid)           { return get("inplay", Map.of("sid", sid)); }
    public String inplayEvent(long id)      { return get("inplayEvent", Map.of("id", id)); }
    public String leagues(int sid)          { return get("leagues", Map.of("sid", sid)); }
    public String prematch(long lid, int sid) {
        return get("prematch", Map.of("lid", lid, "sid", sid));
    }
    public String prematchEvent(long id)    { return get("prematchEvent", Map.of("id", id)); }
    public String upcoming(int sid, String from, String to, String country) {
        Map<String, Object> p = new LinkedHashMap<>();
        p.put("sid", sid); p.put("fromDate", from); p.put("toDate", to);
        if (country != null) p.put("country", country);       // optional
        return get("upcoming", p);
    }
    public String topMatch(String status)   { return get("topMatch", Map.of("status", status)); }
    public String search(String text, String status) {
        return get("search", Map.of("status", status, "text", text));
    }
    public String tournaments()             { return get("tournaments", Map.of()); }
    public String tournamentWinner(long lid) { return get("tournamentWinner", Map.of("lid", lid)); }
    public String resultByEvent(long id)    { return get("resultByEvent", Map.of("id", id)); }

    public static void main(String[] args) {      // usage
        SportsbookClient api = new SportsbookClient();
        System.out.println(api.sports());
        System.out.println(api.inplay(1));
        System.out.println(api.upcoming(1, "2026-10-18", "2026-10-19", "NG"));
        System.out.println(api.search("Premier league", "Line"));
    }
}
```

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**.