

DEVELOPER DOCUMENTATION

Sportsbook API Reference

Sports, live and pre-match events, top matches, leagues and search through one simple, JSON-based HTTP API.

VERSION 3

REST + JSON

13 ENDPOINTS



```
GET https://w7api.com/sports/api/v3/inplay
    ?apikey=0
    &sid=1

200 OK application/json
```

API VERSION

v3

BASE URL

<https://w7api.com/sports/api/v3>

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 events, pre-match (line) events, upcoming matches, top matches, leagues and search. Every endpoint is a simple HTTP **GET** request that returns **JSON**, so it works from any language or platform. This guide documents all 13 endpoints of version 3 and includes ready-to-use client code for **Node.js**, **PHP**, **Kotlin** and **Java**.

API VERSION
v3

RESPONSE FORMAT
JSON

ENDPOINTS
13

AUTH
API key

Contents

Overview	2
Getting Started	3
Endpoint Overview	4
Endpoints	5
/sports - List sports	5
/live - Live events by sport	5
/liveEvent - Live event details	6
/upcoming - Upcoming events by date	6
/lineEvent - Pre-match event details	7
/topLine - Top pre-match matches	7
/topLive - Top live matches	8
/leagueLive - Leagues with live events	8
/leagueLine - Leagues with pre-match events	9
/liveEvents - Live events in a league	9
/lineEvents - Pre-match events in a league	10
/searchLine - Search pre-match events	10
/searchLive - Search live events	11
Code Examples	12
Node.js example	13
PHP example	14
Kotlin example	15
Java example	16
Best Practices & Notes	17

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/v3
```

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/v3/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. In v3 every parameter is required, including `country` on all endpoints except `/sports`.

Typical workflows

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

Browse pre-match events	<code>/sports</code>	,	<code>/leagueLine</code>	,	<code>/lineEvents</code>	,	<code>/lineEvent</code>
Follow live events	<code>/sports</code>	,	<code>/leagueLive</code>	,	<code>/liveEvents</code>	,	<code>/liveEvent</code>
Quick live overview	<code>/live</code>	,	<code>/liveEvent</code>				

REFERENCE

Endpoint Overview

A summary of all 13 endpoints. Click an endpoint in the Contents page for full details.

#	Endpoint	Description	Required parameters
1	/sports	List sports	apikey
2	/live	Live events by sport	apikey, sid, country
3	/liveEvent	Live event details	apikey, id, country
4	/upcoming	Upcoming events by date	apikey, sid, fromDate, toDate, country
5	/lineEvent	Pre-match event details	apikey, id, country
6	/topLine	Top pre-match matches	apikey, sid, country
7	/topLive	Top live matches	apikey, sid, country
8	/leagueLive	Leagues with live events	apikey, sid, country
9	/leagueLine	Leagues with pre-match events	apikey, sid, country
10	/liveEvents	Live events in a league	apikey, sid, lid, country
11	/lineEvents	Pre-match events in a league	apikey, sid, lid, country
12	/searchLine	Search pre-match events	apikey, text, country
13	/searchLive	Search live events	apikey, text, country

Tip: finding IDs

Sport IDs come from [/sports](#) . League IDs come from [/leagueLive](#) and [/leagueLine](#) . Event IDs come from list endpoints such as [/live](#) , [/upcoming](#) , [/liveEvents](#) and [/lineEvents](#) .

New in v3

Live and pre-match (line) data now have their own endpoints, for example [/live](#) , [/liveEvent](#) , [/lineEvent](#) , [/leagueLive](#) and [/searchLine](#) . Every endpoint except [/sports](#) now requires the [country](#) parameter, and the [oddType](#) parameter is not used in v3.

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/v3/sports
?apikey=0
```

02

GET

/live

Live events by sport

Returns live (in-play) events for a sport and country. Use `sid 0` to get all sports.

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	0	Sport ID. Use 0 to return all sports, or a specific sport ID to return only that sport.
country	text	Required	NG	Country code, for example NG.



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v3/live
?apikey=0
&sid=0
&country=NG
```

03

GET

/liveEvent

Live event details

Returns full details of a single live 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	Live event ID, for example one returned by /live .
country	text	Required	NG	Country code, for example NG.



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v3/liveEvent
?apikey=0
&id=1
&country=NG
```

04

GET

/upcoming

Upcoming events by date

Returns upcoming events for a sport within a date range and country.

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	Required	NG	Country code, for example NG.



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v3/upcoming
?apikey=0
&sid=1
&fromDate=2026-10-18
&toDate=2026-10-19
&country=NG
```

05

GET

/lineEvent

Pre-match event details

Returns full details of a single pre-match (line) 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	Pre-match event ID, for example one returned by /upcoming or /lineEvents .
country	text	Required	NG	Country code, for example NG.



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v3/lineEvent
?apikey=0
&id=1
&country=NG
```

06

GET

/topLine

Top pre-match matches

Returns the featured top matches from the pre-match (line) list.

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	0	Sport ID. Use 0 to return all sports, or a specific sport ID to return only that sport.
country	text	Required	NG	Country code, for example NG.



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v3/topLine
?apikey=0
&sid=0
&country=NG
```

07

GET

/topLive

Top live matches

Returns the featured top matches that are currently live.

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	0	Sport ID. Use 0 to return all sports, or a specific sport ID to return only that sport.
country	text	Required	NG	Country code, for example NG.



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v3/topLive
?apikey=0
&sid=0
&country=NG
```

08

GET

/leagueLive

Leagues with live events

Returns the leagues that currently have live events. League IDs are used as lid in [/liveEvents](#).

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	0	Sport ID. Use 0 to return all sports, or a specific sport ID to return only that sport.
country	text	Required	NG	Country code, for example NG.



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v3/leagueLive
?apikey=0
&sid=0
&country=NG
```

09

GET

/leagueLine

Leagues with pre-match events

Returns the leagues that have pre-match (line) events. League IDs are used as `lid` in [/lineEvents](#).

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	0	Sport ID. Use 0 to return all sports, or a specific sport ID to return only that sport.
country	text	Required	NG	Country code, for example NG.



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v3/leagueLine
?apikey=0
&sid=0
&country=NG
```

10

GET

/liveEvents

Live events in a league

Returns the live events of one 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 .
sid	number	Required	-	Sport ID. Get valid IDs from /sports .
lid	number	Required	-	League ID, for example one returned by /leagueLive .
country	text	Required	NG	Country code, for example NG.



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v3/liveEvents
?apikey=0
&sid=1
&lid={leagueId}
&country=NG
```

11

GET

/lineEvents

Pre-match events in a league

Returns the pre-match (line) events of one 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.
sid	number	Required	-	Sport ID. Get valid IDs from /sports .
lid	number	Required	-	League ID, for example one returned by /leagueLine .
country	text	Required	NG	Country code, for example NG.



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v3/lineEvents
?apikey=0
&sid=1
&lid={leagueId}
&country=NG
```

12

GET

/searchLine

Search pre-match events

Searches pre-match (line) 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.
text	text	Required	Premier league	Search keyword. URL-encode spaces (e.g. Premier%20league).
country	text	Required	NG	Country code, for example NG.



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v3/searchLine
?apikey=0
&text=Premier%20league
&country=NG
```

13

GET

/searchLive

Search live events

Searches live 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 .
text	text	Required	Premier league	Search keyword. URL-encode spaces (e.g. Premier%20league).
country	text	Required	NG	Country code, for example NG.



EXAMPLE REQUEST

```
GET https://w7api.com/sports/api/v3/searchLive
?apikey=0
&text=Premier%20league
&country=NG
```

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/v3";
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  live(sid = 0, country = "NG") { return this.#get("live", { sid, country }); }
18  liveEvent(id, country = "NG") { return this.#get("liveEvent", { id, country }); }
19  upcoming(sid, fromDate, toDate, country = "NG") {
20    return this.#get("upcoming", { sid, fromDate, toDate, country });
21  }
22  lineEvent(id, country = "NG") { return this.#get("lineEvent", { id, country }); }
23  topLine(sid = 0, country = "NG") { return this.#get("topLine", { sid, country }); }
24  topLive(sid = 0, country = "NG") { return this.#get("topLive", { sid, country }); }
25  leagueLive(sid = 0, country = "NG") { return this.#get("leagueLive", { sid, country }); }
26  leagueLine(sid = 0, country = "NG") { return this.#get("leagueLine", { sid, country }); }
27  liveEvents(sid, lid, country = "NG") {
28    return this.#get("liveEvents", { sid, lid, country });
29  }
30  lineEvents(sid, lid, country = "NG") {
31    return this.#get("lineEvents", { sid, lid, country });
32  }
33  searchLine(text, country = "NG") { return this.#get("searchLine", { text, country }); }
34  searchLive(text, country = "NG") { return this.#get("searchLive", { text, country }); }
35 }
36
37 // ----- usage -----
38 (async () => {
39   const api = new SportsbookClient("0");
40   console.log(await api.sports());
41   console.log(await api.live(0, "NG"));
42   console.log(await api.upcoming(1, "2026-10-18", "2026-10-19", "NG"));
43   console.log(await api.searchLine("Premier league", "NG"));
44 })().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/v3';
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 live(int $sid = 0, string $country = 'NG'): array
34     { return $this->get('live', compact('sid', 'country')); }
35     public function liveEvent(int $id, string $country = 'NG'): array
36     { return $this->get('liveEvent', compact('id', 'country')); }
37     public function upcoming(int $sid, string $fromDate, string $toDate, string $country = 'NG'): array
38     { return $this->get('upcoming', compact('sid', 'fromDate', 'toDate', 'country')); }
39     public function lineEvent(int $id, string $country = 'NG'): array
40     { return $this->get('lineEvent', compact('id', 'country')); }
41     public function topLine(int $sid = 0, string $country = 'NG'): array
42     { return $this->get('topLine', compact('sid', 'country')); }
43     public function topLive(int $sid = 0, string $country = 'NG'): array
44     { return $this->get('topLive', compact('sid', 'country')); }
45     public function leagueLive(int $sid = 0, string $country = 'NG'): array
46     { return $this->get('leagueLive', compact('sid', 'country')); }
47     public function leagueLine(int $sid = 0, string $country = 'NG'): array
48     { return $this->get('leagueLine', compact('sid', 'country')); }
49     public function liveEvents(int $sid, int $lid, string $country = 'NG'): array
50     { return $this->get('liveEvents', compact('sid', 'lid', 'country')); }
51     public function lineEvents(int $sid, int $lid, string $country = 'NG'): array
52     { return $this->get('lineEvents', compact('sid', 'lid', 'country')); }
53     public function searchLine(string $text, string $country = 'NG'): array
54     { return $this->get('searchLine', compact('text', 'country')); }
55     public function searchLive(string $text, string $country = 'NG'): array
56     { return $this->get('searchLive', compact('text', 'country')); }
57 }
58
59 // ----- usage -----
60 $api = new SportsbookClient('0');
61 print_r($api->sports());
62 print_r($api->live(0, 'NG'));
63 print_r($api->upcoming(1, '2026-10-18', '2026-10-19', 'NG'));
64 print_r($api->searchLine('Premier league', 'NG'));

```

CODE EXAMPLE

Kotlin

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

Run: Run main() from your IDE or Gradle.

```

KOTLIN - SportsbookClient

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/v3"
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 live(sid: Int = 0, country: String = "NG") =
29         get("live", mapOf("sid" to sid, "country" to country))
30     fun liveEvent(id: Long, country: String = "NG") =
31         get("liveEvent", mapOf("id" to id, "country" to country))
32     fun upcoming(sid: Int, fromDate: String, toDate: String, country: String = "NG") =
33         get("upcoming", mapOf("sid" to sid, "fromDate" to fromDate,
34             "toDate" to toDate, "country" to country))
35     fun lineEvent(id: Long, country: String = "NG") =
36         get("lineEvent", mapOf("id" to id, "country" to country))
37     fun topLine(sid: Int = 0, country: String = "NG") =
38         get("topLine", mapOf("sid" to sid, "country" to country))
39     fun topLive(sid: Int = 0, country: String = "NG") =
40         get("topLive", mapOf("sid" to sid, "country" to country))
41     fun leagueLive(sid: Int = 0, country: String = "NG") =
42         get("leagueLive", mapOf("sid" to sid, "country" to country))
43     fun leagueLine(sid: Int = 0, country: String = "NG") =
44         get("leagueLine", mapOf("sid" to sid, "country" to country))
45     fun liveEvents(sid: Int, lid: Long, country: String = "NG") =
46         get("liveEvents", mapOf("sid" to sid, "lid" to lid, "country" to country))
47     fun lineEvents(sid: Int, lid: Long, country: String = "NG") =
48         get("lineEvents", mapOf("sid" to sid, "lid" to lid, "country" to country))
49     fun searchLine(text: String, country: String = "NG") =
50         get("searchLine", mapOf("text" to text, "country" to country))
51     fun searchLive(text: String, country: String = "NG") =
52         get("searchLive", mapOf("text" to text, "country" to country))
53 }
54
55 // ----- usage -----
56 fun main() {
57     val api = SportsbookClient()
58     println(api.sports())
59     println(api.live(0, "NG"))
60     println(api.upcoming(1, "2026-10-18", "2026-10-19", "NG"))
61     println(api.searchLine("Premier league", "NG"))
62 }

```

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/v3";
11     private final HttpClient http = HttpClient.newBuilder()
12         .connectTimeout(Duration.ofSeconds(10)).build();
13     private final String apiKey;
14     public SportsbookClient() { this("0"); } // "0" = 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, Object... kv) { // kv = name, value, name, value ...
20         StringJoiner query = new StringJoiner("&");
21         query.add("apikey=" + enc(apiKey));
22         for (int i = 0; i < kv.length; i += 2)
23             query.add(kv[i] + "=" + enc(String.valueOf(kv[i + 1])));
24         try {
25             HttpRequest req = HttpRequest
26                 .newBuilder(URI.create(BASE_URL + "/" + endpoint + "?" + query))
27                 .timeout(Duration.ofSeconds(15)).GET().build();
28             HttpResponse<String> res = http.send(req, HttpResponse.BodyHandlers.ofString());
29             if (res.statusCode() != 200) throw new IOException("HTTP " + res.statusCode());
30             return res.body();
31         } catch (IOException | InterruptedException e) {
32             throw new RuntimeException("Request failed: /" + endpoint, e);
33         }
34     }
35     public String sports() { return get("sports"); }
36     public String live(int sid, String cc) { return get("live", "sid", sid, "country", cc); }
37     public String liveEvent(long id, String cc) { return get("liveEvent", "id", id, "country", cc); }
38     public String upcoming(int sid, String from, String to, String cc) {
39         return get("upcoming", "sid", sid, "fromDate", from, "toDate", to, "country", cc);
40     }
41     public String lineEvent(long id, String cc) { return get("lineEvent", "id", id, "country", cc); }
42     public String topLine(int sid, String cc) { return get("topLine", "sid", sid, "country", cc); }
43     public String topLive(int sid, String cc) { return get("topLive", "sid", sid, "country", cc); }
44     public String leagueLive(int sid, String cc) { return get("leagueLive", "sid", sid, "country", cc); }
45     public String leagueLine(int sid, String cc) { return get("leagueLine", "sid", sid, "country", cc); }
46     public String liveEvents(int sid, long lid, String cc) {
47         return get("liveEvents", "sid", sid, "lid", lid, "country", cc);
48     }
49     public String lineEvents(int sid, long lid, String cc) {
50         return get("lineEvents", "sid", sid, "lid", lid, "country", cc);
51     }
52     public String searchLine(String q, String cc) { return get("searchLine", "text", q, "country", cc); }
53     public String searchLive(String q, String cc) { return get("searchLive", "text", q, "country", cc); }
54
55     public static void main(String[] args) { // usage
56         SportsbookClient api = new SportsbookClient();
57         System.out.println(api.sports());
58         System.out.println(api.live(0, "NG"));
59         System.out.println(api.upcoming(1, "2026-10-18", "2026-10-19", "NG"));
60         System.out.println(api.searchLine("Premier league", "NG"));
61     }
62 }

```

GUIDANCE

Best Practices & Notes

- 1 Cache slow-changing lists**
The sports list from `/sports` changes rarely. Cache it, and cache league lists from `/leagueLive` and `/leagueLine` for a short time instead of requesting them on every page load.
- 2 Poll live data sensibly**
For `/live`, `/liveEvent` and `/topLive` 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 Always send the country**
Every endpoint except `/sports` requires the `country` parameter, for example `NG`.
- 6 Use sid 0 on purpose**
On `/live`, `/topLine`, `/topLive`, `/leagueLive` and `/leagueLine`, `sid=0` returns all sports. Pass a specific sport ID to receive only that sport.

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.