

Bodkin Boats API Documentation

Base URL: `https://files.cablepost.co.uk/bodkinBoats/`

All endpoints return JSON and accept parameters via query string. No authentication is required.

Endpoints

1. Get Seasons

Returns a list of all available season IDs.

GET `/get_seasons.php`

Parameters: None

Example Request:

```
GET https://files.cablepost.co.uk/bodkinBoats/get_seasons.php
```

Example Response:

```
["test_season"]
```

The response is a plain JSON array of season ID strings.

2. Get Season Standings

Returns the championship standings for a given season, sorted by points (descending).

GET `/get_season_standings.php`

Parameter	Type	Required	Description
<code>season</code>	string	Yes	The season ID (from <code>get_seasons</code>)

Example Request:

```
GET https://files.cablepost.co.uk/bodkinBoats/get_season_standings.php?season=test_season
```

Example Response:

```
[
  {
    "uuid": "a6c86c96-06c9-4585-ae88-c8bede08c453",
    "name": "EchoEvo",
    "points": 630
  },
  {
    "uuid": "d2fa6c66-97d4-440f-b90f-e7b2f8355512",
    "name": "AvatarAa",
    "points": 608
  }
]
```

Response Fields:

Field	Type	Description
uuid	string	Unique identifier for the player
name	string	Player's display name
points	integer	Total championship points this season

3. Get Season Races

Returns a list of all races in a given season, in chronological order.

GET /get_season_races.php

Parameter	Type	Required	Description
season	string	Yes	The season ID (from <code>get_seasons</code>)

Example Request:

```
GET https://files.cablepost.co.uk/bodkinBoats/get_season_races.php?
season=test_season
```

Example Response:

```
[
  {
    "name": "test_event_1_race_1",
    "happened": true,
    "timestamp": 1732986057288
  },
  {
    "name": "test_event_1_race_2",
    "happened": true,
    "timestamp": 1732987144289
  }
]
```

Response Fields:

Field	Type	Description
name	string	Unique race identifier, used as the <code>race</code> parameter in <code>get_race</code>
happened	boolean	Whether the race has taken place
timestamp	integer	Unix timestamp (milliseconds) of when the race started

4. Get Race

Returns the full data for a specific race, including the starting grid, checkpoint-by-checkpoint tracking data, and final results.

GET `/get_race.php`

Parameter	Type	Required	Description
season	string	Yes	The season ID (from <code>get_seasons</code>)
race	string	Yes	The race name (from <code>get_season_races</code>)

Example Request:

```
GET https://files.cablepost.co.uk/bodkinBoats/get_race.php?
season=test_season&race=test_event_1_race_1
```

Example Response (abbreviated):

```

{
  "grid": [
    { "name": "BillBodkin", "uuid": "3896d814-d160-4cb1-9dd6-bc0f180e59e1" },
    { "name": "Power_Blackflame", "uuid": "def5bd28-54f4-4271-a6f2-98f0fa57fbbc" }
  ],
  "goTimestamp": 1732986057288,
  "maxLaps": 5,
  "checkpoints": [
    {
      "uuid": "d2fa6c66-97d4-440f-b90f-e7b2f8355512",
      "name": "AvatarAa",
      "lap": 1,
      "checkpoint": 1,
      "timestamp": 1732986065710
    }
  ],
  "players": [
    {
      "name": "BillBodkin",
      "uuid": "3896d814-d160-4cb1-9dd6-bc0f180e59e1",
      "gridPosition": 1,
      "finishPosition": 1,
      "finishPositionStr": "1",
      "points": 50,
      "fastestLap": 116550,
      "fastestLapStr": "01:56.550"
    },
    {
      "name": "Power_Blackflame",
      "uuid": "def5bd28-54f4-4271-a6f2-98f0fa57fbbc",
      "gridPosition": 2,
      "finishPosition": -1,
      "finishPositionStr": "DNF",
      "points": 0,
      "fastestLap": 118445,
      "fastestLapStr": "01:58.445"
    }
  ]
}

```

Top-level Response Fields:

Field	Type	Description
grid	array	Starting grid, ordered by grid position (position 1 first)
goTimestamp	integer	Unix timestamp (milliseconds) when the race started
maxLaps	integer	Total number of laps in the race
checkpoints	array	Full chronological log of every checkpoint crossing
players	array	Final results summary for each participant

grid Object Fields:

Field	Type	Description
name	string	Player's display name
uuid	string	Player's unique identifier

checkpoints Object Fields:

Field	Type	Description
uuid	string	UUID of the player who crossed the checkpoint
name	string	Display name of the player
lap	integer	Lap number on which the checkpoint was crossed
checkpoint	integer	Checkpoint number within that lap
timestamp	integer	Unix timestamp (milliseconds) of the crossing

players Object Fields:

Field	Type	Description
<code>name</code>	string	Player's display name
<code>uuid</code>	string	Player's unique identifier
<code>gridPosition</code>	integer	Starting position on the grid
<code>finishPosition</code>	integer	Final finishing position, or <code>-1</code> for DNF
<code>finishPositionStr</code>	string/integer	Human-readable finish position (<code>"1"</code> , <code>"2"</code> , etc.) or <code>"DNF"</code>
<code>points</code>	integer	Championship points awarded for this race (0 for DNF)
<code>fastestLap</code>	integer	Player's fastest lap time in milliseconds, or <code>-1</code> if unavailable
<code>fastestLapStr</code>	string	Human-readable fastest lap time (e.g. <code>"01:56.550"</code>) or <code>"N/A"</code>

Typical Usage Flow

A typical workflow to display a race result looks like this:

1. Call `get_seasons` to retrieve the list of available seasons.
2. Call `get_season_races` with your chosen season to get the list of races.
3. Call `get_race` with your chosen season and race to get the full race data.
4. Optionally call `get_season_standings` to show the overall championship table.

```
// Example: fetch and display a race result
const season = "test_season";
const race = "test_event_1_race_1";

const response = await fetch(
  `https://files.cablepost.co.uk/bodkinBoats/get_race.php?
season=${season}&race=${race}`
);
const data = await response.json();

// Show the podium
data.players
  .filter(p => p.finishPosition > 0)
  .sort((a, b) => a.finishPosition - b.finishPosition)
  .forEach(p => {
    console.log(`P${p.finishPositionStr} - ${p.name} (${p.fastestLapStr})`);
  });
```