



Podium Racing API Onboarding Documentation

Podium Sports API

Welcome to the Podium Sports API. This document serves as an introduction to our API, and a complement to our [Stoplight Documentation](#) which provides technical information for developers on how to integrate.

The Racing API offers all of Podium's card, betting and results data via a RESTful API. For data where time is of the essence we also offer the ability to receive PUSH deliveries of raceday specific data points.

Products and Services:

The following services are available from the Podium API.

Racecards:

We will make any information relating to future events available in the API as soon as we receive it from the relevant authority. This can be weeks ahead, and for some races, months. Typically, we would expect card data for UK and Irish racing to be present from the six-day stage ahead of the 48-hour declarations. Note, a full breakdown of the data points in the API will be provided in the appendix.

The API will provide global racing coverage across UK, USA, France, South Africa, amongst others. Please see Appendix A for a full list of countries and tracks covered in the API.

Card : The API will offer a full outline of upcoming meetings, including venues, race times, going, horses, jockeys, and supplemental information on trainers, owners and breeders.

Editorial previews : The API offers curated commentary on both races and the horses participating in them (verdicts and pinstickers) as well as tips.



Forecast Price: Podium will advertise a forecast price in advance of shows being available, up to two days ahead of race start.

Form : We offer form information for horses, jockeys and trainers, including career information. We also offer a detailed account via dedicated endpoint of the last six races a horse has participated in.

Silks : Podium will provide access to our Bombyx Silks generator via the API using a download associated directly to the runner. This offers high quality SVG images of the jockey silks used in a race. Please note that you will need to be whitelisted in order to use this service, so we will need the IP address(es) of where you will be accessing the silks from if not in the UK.

Raceday Data

Race Status : Podium will notify you of the current state of the race from 'going down', 'going behind' etc. through 'off' all the way to 'weighed-in'. We will also notify you any post-race challenges or amendments to the result.

Betting: On Raceday, Podium will publish live price updates for all participating horses in a race from the Opening Show through to the SP. We will update prices in the run up to the off, and once the race has started will immediately confirm the SP for the horse. Betting shows will typically start to appear c. 15 minutes prior to the start of the race.

Results : On completion of a race, you will be able to access results. Again, based on permissions and territory, this will also include close-up comments once available.

Horse and jockey changes : Any changes to runners and riders will be published as soon as it is available.

Commentary : Post race commentary (close up) will be available once populated

API structure

Our API has a fully documented Stoplight which contains detailed information on the structure of the endpoints, the format of the data and filters used on any given endpoint to restrict data. This is available here <https://pa-betting-services.stoplight.io>.



Our API is designed as a hierarchical structure, with our list stages endpoint at the top of the hierarchy

- I can retrieve a list of events (races) and associated data (i.e. going, prizes, status) from the stages (meetings) endpoint
- I can retrieve a list of entrants (horses) and associated data (i.e. betting, results, form, etc.) from the events (races) endpoint
- I can retrieve a list of participants (trainers, jockeys, and more horse information) from the entrants (horses) endpoint

However, once you have a stage id, you can also retrieve everything under stages from the stage details endpoint. These are extremely verbose responses (over 20k lines for a meeting) so they need to be parsed to find the data you need. Response times can therefore be impacted due to the volume of data.

(This structure also applies for events, entrants and participants)

- Stages : The stages endpoint provides a high level summary of all race meetings happening within a specific data window. This should be configured using the 'from' and 'to' fields in the API request. We also offer the ability to return a particular country using the 'countrycodes' filter (all codes available from the countries endpoint), and the ability to search for particular courses with the 'query' term.
 - Your response will return a stage id, venue information, and a list of events (races) at each stage (meeting).
 - We also include some summary metadata for each event.
- Events: The Events endpoint is designed to return specific information about all races in the supplied date range. This data will not include Stage information (i.e. venue), but will include the name of the race, the status of the race, the prize money available for the race and the time the race will start. We also carry information on the class of the race, the going, whether it is jumps or flat, distance, and surface type.
 - It contains a list of entrants, their stall and cloth number, and their status as an active entry in the race or a non-runner.
 - It contains the silk for each horse.



- Once complete the event will also be enhanced with timestamps for status changes during the race (i.e. the off time, weighed in time)

To obtain the following data sets you should interrogate the **DETAILS** endpoint

- Race dividends (CSF, tricasts, tote, exacta, trifecta)
- Tips
- Prize money
- Additional content (race comments, Timeform comments, Timeform Smart stats)

We also offer a view of race renewals such that the user can return previous winners

- For example, the Qipco Sprint Stakes at Ascot in 2024 has an event uuid of 291d3994-f3ca-4774-8fc2-17802505c083 .
 - This can be queried on `{{ $baseURL }}/events/291d3994-f3ca-4774-8fc2-17802505c083/winners` which returns Art Power as the winner in 2023.
- Entrants: Entrants requires that you return a given event uuid, and then will return all of the horses participating in that event. So for an example UUID 73751e7f-4edf-4226-b1bf-62c16b0a6e0d (Fairyhouse, IRE, 1st December 2024) would return a list of the horses in that event, supporting information and high level participant information on the jockey, trainer and owner. You will also obtain
 - Ratings (official and Timeform)
 - All odds generated for the entrant over the course of the race
 - Assorted metadata for the entrant, including distance travelled, form string, weight, days since last run
 - Finish position or status if unfinished (ie fallen, disqualified, unseated rider, etc.)
 - High level participant data such as jockey, trainer, owner, breeder
 - Participant data for the horse, such as age, career history, gender, colour and prize money.
 - Lineage
- Entrant form - to see (up to) the last six races a horse has competed in you can use the entrants endpoint with the qualifier /form to pull previous results.
 - So for entrant uuid f5158412-7356-42a1-ac5d-763b787657a2 (End Zone) it's previous races can be obtained by sending `{{ $baseURL }}/entrants/f5158412-7356-42a1-ac5d-763b787657a2/form`



- Participants : Participants covers all horse related entities. So in this case, participants will also include information on the jockey, owner and trainer, as well as the horse itself. This data should be duplicated on the entrant details endpoint.

Revisions

We have introduced the concept of 'history' endpoints into our build to allow you to examine the update history of a venue, stage, event, entrant or participant. Every update we make to an entity in our DB is stored and available to return.

- The format for any query is {{base URL}}/ ENDPOINT / uuid / history
- So, for a given stage, the format is /stages/uuid/history
 - This will provide a full history of every data update for a meeting. Note this can be sizeable response. Leicester on the 1st December 2024 is 165k lines of JSON, and 400KB.
- For a given event, events/uuid/history
 - this is useful for seeing changes in status, or betting updates, or the specific off update
- For a given entrant, entrants/uuid/history (note this is distinct from form)
 - this is useful for entrant specific odds checks
- For a participant participants/uuid/history (this should update after each race)
 - this could be used to generate a race by race career history for the horse or jockey

Minor Endpoints

We also carry some minor endpoints exposing datasets used in other responses. We do not anticipate you needing to call these on a regular basis.

- Country : A list of all countries, their timezones, and their respective currencies. Note that because a country appears in this list does not mean we are carrying racing information. The main use case for this would be to obtain the code needed for country filtering in the stages endpoint.
- Currency : Information on all currencies present in the API
- Surface: This endpoint provides a list of all types of surface present in the API
- Sport : At present this just covers horse racing.



Note that the response on all endpoints can be filtered to return only the data points you are interested in. This is explained in far greater detail in Appendix D - fields filters

Connecting to the API

We have a POSTMAN collection available to simplify making calls to the API. If this has not been provided, it can be requested. We can also make available an environment file with your credentials pre-populated.

Access to the API is granted via the 'get token' request which allows access for 30 minutes. The API operates on basic auth (in Postman - select a call, select authorisation, select basic auth) - you will need to input the username / password credentials you have been supplied with by Podium. Note that you can set these as defaults using the environment tab to pre-populate (and mask) access credentials

Note that the API has a return limit of 6MB. If you exceed this you will be notified of an internal server error. Use the pagesize filter to limit the page return to 1 and see if that solves the problem.

Caching

To optimize API performance and reduce the load on our infrastructure, we employ caching mechanisms. Through Amazon ElastiCache, we store frequently requested data in memory, allowing faster retrieval and reducing the need to fetch the same data repeatedly. This caching strategy enhances the overall responsiveness and scalability of our API.

We expect cached responses to take around 40 - 60 ms to return

Uncached responses can take up to several seconds depending on the size of the response.

Our API supports responses up to 6MB in response.



Technical Support and Documentation

As previously stated, we have a Stoplight for helping to guide you through the API
<https://pa-betting-services.stoplight.io>

In the event that the API is not working, or there are issues with the data that is being received, please contact:

Steve Whitehead

- Email: Steve.whitehead@podiumsports.com

and

William Gibson

- Email: William.gibson@pamediaigroup.com

We hope this introduction document helps you get started in using the Podium API. We are excited to have you onboard and look forward to working with you on your horse racing integration. In the event of any problems, don't hesitate to get in touch as we are keen to understand your experience of the API.



Appendix A – Country list

Countries available in the API

UK

Ireland

Japan

USA

France

South Africa

Chile

Brazil

India

Hong Kong

Saudi Arabia

Uruguay



Appendix B – PUSH data

For time critical services, our PUSH data allows small snippets of data to be provided to customers to integrate with their general data set.

The following data points are made available via PUSH delivery

Event : Conditions (weather conditions for the race, rather than the meeting)

Event : Dividends (tote returns, CSF, tricast, trifecta, etc)

Entrant : Bet movements (advice on either big bets, or commentary on bet movement, such as 'opened at 28/1)

Entrant : Close-up (editorial commentary post race describing a horses performance)

Entrant : Jockey Weight (the weight allowance claimed by the jockey)

Entrant : Non-Runner (a notification that a horse is now a non runner)

Entrant : Reinstate (a non runner has been reinstated and now will race)

Entrant : Result (the horses finish position)

Entrant : Shows (updates to the horses odds)

Entrant : Starting Prices (confirmation of the price of the horse at the off)

Entrant : Withdrawn (a horse has been withdrawn from the race)

Event : Deduction (a notification that returns will be reduced following a withdrawal or a change to result)

Event : Market Resumed (a bet market has been reopened)

Event : Market Suspended (a bet market has been suspended)

Event : Off Time Correction (an updated offtime has been sent)



Event : Status (a change to the status of the race, e.g. from dormant, to going down, to off, to finished)

Event : Stewards (there is a stewards enquiry into the outcome of the race)

Event : Winning Time (the time the winner of the race took from start to finish)

Stage : Conditions (the going conditions of the meeting - the weather and the state of the ground)

Stage : Dividends (dividends relating to meeting level outcomes, ie jackpot)

Stage : Status (the status of the meeting)



Appendix C – PUSH setup

The PUSH service is designed to complement the rest API. On Raceday, a user can elect to receive updates via our AWS SNS PUSH delivery service.

This is designed to maximise the speed of generation and delivery of critical race day data.

Whilst a user could elect to poll the API several times per second in order to keep their data as fresh as possible, PUSH offers the ability to just receive the core data updates rather than parsing longer data strings.

It is important to note that a user will need to retrieve the card data from the REST API. We do not offer full card data over PUSH in order to minimise the data payloads and latency.

More information on message formats is available in Annex B, but it is important to note that we will send you the updated (or new) data value, and a sufficient amount of reference data to link the data to the relevant stage, event, or entrant. We do not send the previous value.

At any point during Raceday a customer may repoll the REST API in order to update their data set, or re-sync with Podium's formal version of the truth. Every PUSH update is mirrored in the dataset within the REST API, and via the use of revision endpoints it is possible to see when an update was generated

Setup

The PUSH service setup is automated on the Podium side. When we enable you for PUSH, we trigger an AWS stack creation process resulting in an SNS for you to connect to, where you can retrieve messages. This SNS will reflect your contractual entitlements to data, so for example, if you take GB data but not French, you will only receive relevant GB data on the SNS.

We require your AWS IAM to enable you to access the environment we create. This is stored securely within our entitlements DB.



PUSH data stubs example SHOWS

Our PUSH services sends minimal data elements which are designed to update elements of the full data set you receive

Betting shows

```
"body": "[{"correlationId":"4afbeb86-4aa6-483a-937a-41d4a1ad6ed9","features":{"BETTING_SHOWS"},"message":{"id":"4afa5384-b4bc-4951-9ad3-1d35a6742cbc","type":"event","entrants":[{"id":"30fb7908-3208-48bb-9b59-f9da8b9480ed","type":"entrant","betting":{"shows":{"marketnumber":1,"price":{"decimal":200,"denominator":1,"numerator":200},"created":"2024-12-12T17:50:46.618Z"}}}]"}]
```

In this example

note the message type is declared as BETTING_SHOWS in the features element.

The first relevant piece of information in the message is

```
"message":{"id":"4afa5384-b4bc-4951-9ad3-1d35a6742cbc","type":"event"
```

This tell us the UUID of the event that is being referenced, in this case 4afa5384-b4bc-4951-9ad3-1d35a6742cbc.

```
{{baseURL}}/events/4afa5384-b4bc-4951-9ad3-1d35a6742cbc)
```

which is the 18:00 at Chelmsford City on 12th December 2025.

The next relevant piece of information is

```
"entrants":[{"id":"30fb7908-3208-48bb-9b59-f9da8b9480ed"
```

```
{{baseURL}}/entrants/30fb7908-3208-48bb-9b59-f9da8b9480ed)
```

This tell us that the horse being returned is Pink Petunia in the race above.

- Note than at update can cover more than one entrant

The final key piece of information is the actual payload for the entrant, which is



```
\\\"betting\\\":{\\\"shows\\\":[{\\\"marketnumber\\\":1,\\\"price\\\":{\\\"decimal\\\":200,\\\"denominator\\\":1,\\\"numerator\\\":200}
```

Here you can see that we've sent a betting update, and the new shows (both decimal and fractional) are available on the PUSH message.

Where multiple entrants are sent, data is sent in an entrant / show / entrant / show format as in this example

```
"body": "[{\\\"correlationId\\\":\\\"18d28297-0211-4d4b-ac3f-ae70ab8e7b74\\\",\\\"features\\\":[\\\"BETTING_SHOWS\\\"],\\\"message\\\":\\\"{\\\"id\\\":\\\"646f1b6a-f681-4334-b54c-c2f50b7a144f\\\",\\\"type\\\":\\\"event\\\",\\\"entrants\\\":[{\\\"id\\\":\\\"e3304cb4-9158-4b49-8e78-cacd60e8b370\\\",\\\"type\\\":\\\"entrant\\\",\\\"betting\\\":{\\\"shows\\\":[{\\\"marketnumber\\\":1,\\\"price\\\":{\\\"decimal\\\":28,\\\"denominator\\\":1,\\\"numerator\\\":28},\\\"created\\\":\\\"2024-12-12T17:20:33.137Z\\\"}]}],{\\\"id\\\":\\\"11bae700-c6e5-4987-a558-67fac5e503b3\\\",\\\"type\\\":\\\"entrant\\\",\\\"betting\\\":{\\\"shows\\\":[{\\\"marketnumber\\\":1,\\\"price\\\":{\\\"decimal\\\":150,\\\"denominator\\\":1,\\\"numerator\\\":150},\\\"created\\\":\\\"2024-12-12T17:20:33.137Z\\\"}]}]}]\"}
```

PUSH data stubs example - STATUS

This is an example of a status change, in this saying the race has finished.

```
"[\\\"correlationId\\\":\\\"76b23408-2e0e-4612-b8f6-7c7116a27dbb\\\",\\\"features\\\":[ ],\\\"message\\\":\\\"{\\\"id\\\":\\\"c4c30487-1ee9-4864-886a-11516985cefe\\\",\\\"type\\\":\\\"event\\\",\\\"finished\\\":\\\"2024-12-12T17:03:32.995Z\\\",\\\"status\\\":\\\"finished\\\",\\\"meta\\\":{\\\"event:status:finished:timestamp\\\":{\\\"code\\\":\\\"event:status:finished:timestamp\\\",\\\"value\\\":\\\"2024-12-12T17:03:32.995Z\\\"}}}\"}
```

For mapping, we provide the event id of

c4c30487-1ee9-4864-886a-11516985cefe

which is the 5:00pm at Chelmsford on 12th December 2025.

This message indicates that the race was marked as finished and includes a timestamp of when that occurred.



This is the off for the 6:00pm at Chelmsford, which is in an identical format, except for the fact the status is 'off', and the associated timestamp is differently named.

```
[{"correlationId":"23b2508d-26a7-463a-8458-405c4b206830","features":[],"message":{"id":"4afa5384-b4bc-4951-9ad3-1d35a6742cbc","type":"event","offtime":"2024-12-12T18:01:17.117Z","status":"off","meta":{"event:status:off:timestamp":{"code":"event:status:off:timestamp","value":"2024-12-12T18:01:17.117Z"}}}]}
```

PAGE data stubs example - GOING / WEATHER

This example shows an update to the weather / going of an event.

```
[{"correlationId":"7ef44900-2c46-48dd-b289-4096bb0b7437","features":[],"message":{"id":"c4c30487-1ee9-4864-886a-11516985cefe","type":"event","meta":{"event:going":{"code":"event:going","value":"standard"},"event:weather":{"code":"event:weather","value":"Light Rain Showers"}}}]}
```

In this case the event ID is c4c30487-1ee9-4864-886a-11516985cefe

which is the 5pm at Chelmsford on the 12th December 2024.

This messages says that the event going has been changed to standard, and the weather updated to Light Rain Showers.



Appendix D – fields filters

These features are more thoroughly documented within our stoplight (<https://pa-betting-services-stoplight.io>) under the query parameters section of each endpoint, but the API offers the ability to return subsets of the fields available on an endpoint, or to search for specific data strings within the API.

For example, on the venue endpoint, you may only be interested in the IDs of the tracks and their name. So, setting a 'fields' query in POSTMAN to 'id,name' (you don't need the speech marks) would return a list of all the venues in the system, with their UUID, and their name.

```
{{ $baseUrl }}/venues?fields=id,name
```

This will return a list of every venue in the system.

Say however, that you want to see the country that the track belongs to, and you extend the fields list to include that (`{{ $baseUrl }}/venues?fields=id,name,country`) = you will notice that we are returning ALL of the data under the country endpoint, which is of limited use.

To limit the return of data on a nested endpoint you need to ensure you've got URL encoding enabled, and then submit the following string

```
{{ $baseUrl }}/venues?fields=id,name,country{name,code}
```

and this will return the name of the track, the id of the track, the country where the track is and where the track is located

Say you want to filter on tracks in a certain territory? At present you can search on the country code or the countryid (name filtering is not currently available)



```
{{ $baseUrl }}/venues?fields=id,name,country{name,code}&countrycodes=UK
```

This query will return the id and name of all tracks stored in the system for the UK

```
{{ $baseUrl }}/venues?fields=id,name,country{name,code,timezones,id}&countryids=b18b2ec5-f60e-4255-bf12-5c258c223d5d&pagesize=25
```

So this query will return the id and name from the venue, as well as the name,code,timezone and ID for the ID in the query (which happens to be England). This also sets the number of results returned on the query to 25 rather than the default 10.

STAGES - Fields filters examples

Stages endpoint

- I want to simply see the venue, the race names, and the start time

```
{{ $baseUrl }}/stages?fields=id,name,start,end,events[ ][id,title,start]&from=2024-12-04&to=2024-12-04&countrycodes=UK&pagesize=10
```

returns (this is a partial response for brevity)

```
{
  "pagesize": 10,
  "total": 4,
  "filters": {
    "from": "2024-12-04",
    "to": "2024-12-04",
    "fields": "id,name,start,end,events[ ][id,title,start]",
    "countrycodes": [
      "UK"
    ]
  }
}
```



```

},
"items": [
  {
    "id": "50adf989-535e-434d-b32a-e8d4c86d0252",
    "name": "Lingfield",
    "start": "2024-12-04T11:20:00.000Z",
    "end": "2024-12-04T15:25:00.000Z",
    "events": [
      {
        "id": "80335f1d-0264-4339-88d1-15f7c643d799",
        "title": "Win 1M With BetMGM's Golden Goals Handicap",
        "start": "2024-12-04T11:20:00.000Z"
      },
      {
        "id": "c6b595d4-efbd-40d5-a635-446e9f0cf7e6",
        "title": "BetMGM: It's Showtime Nursery",
        "start": "2024-12-04T11:50:00.000Z"
      },
      {
        "id": "5b2fd2f7-946e-41ea-a33c-a28f0ec7f1d9",
        "title": "Gamble Responsibly At BetMGM/ EBF Fillies' Novice Stakes (GBB Race)(Div 1)",
        "start": "2024-12-04T12:20:00.000Z"
      }
    ]
  }
]

```

- I want to see the stage information and it's associated metadata (such as going, weather and draw advantage)
 - *the information returned on the meta stub is not available for filtering.*

{{ \$baseUrl }}/stages?from=2024-12-04&to=2024-12-04&countrycodes=UK&pagesize=10&fields=id,name,start,end,meta

returns this response

```

"filters": {
  "from": "2024-12-04",
  "to": "2024-12-04",
  "fields": "id,name,start,end,meta",
  "countrycodes": [
    "UK"
  ]
},
"items": [
  {
    "id": "50adf989-535e-434d-b32a-e8d4c86d0252",
    "name": "Lingfield",

```



```
"start": "2024-12-04T11:20:00.000Z",
"end": "2024-12-04T15:25:00.000Z",
"meta": {
  "stage:going": {
    "code": "stage:going",
    "value": "standard"
  },
  "stage:weather": {
    "code": "stage:weather",
    "value": "Mostly Cloudy"
  },
  "stage:drawadvantage": {
    "code": "stage:drawadvantage",
    "value": "Little effect."
  },
  "stage:going:description": {
    "code": "stage:going:description",
    "value": null
  }
}
```

- I want to see a list of the races and their current status, and their off and finished times

```
{{${baseUrl}}}/stages?from=2024-12-04&to=2024-12-04&countrycodes=UK&pagesize=10&query=Kempton&fields=id,name,start,end,events[ ]{start,status,finished, offtime}
"items": [
  {
    "id": "abd95869-59cc-496e-b9d6-bb48520b9fa6",
    "name": "Kempton",
    "start": "2024-12-04T16:10:00.000Z",
    "end": "2024-12-04T20:10:00.000Z",
    "events": [
      {
        "start": "2024-12-04T16:10:00.000Z",
        "status": "HORSE_RACING_EVENT_WEIGHED_IN",
        "finished": "2024-12-04T16:11:58.000Z",
        "offtime": "2024-12-04T16:10:29.000Z"
      },
      {
        "start": "2024-12-04T16:40:00.000Z",
        "status": "HORSE_RACING_EVENT_WEIGHED_IN",
        "finished": "2024-12-04T16:45:00.000Z",
        "offtime": "2024-12-04T16:43:46.000Z"
      }
    ]
  }
]
```



```

},
{
  "start": "2024-12-04T17:10:00.000Z",
  "status": "HORSE_RACING_EVENT_WEIGHED_IN",
  "finished": "2024-12-04T17:12:44.000Z",
  "offtime": "2024-12-04T17:11:30.000Z"
},
{
  "start": "2024-12-04T17:40:00.000Z",
  "status": "EVENT_RESULT",
  "finished": "2024-12-04T17:43:56.000Z",
  "offtime": "2024-12-04T17:43:56.000Z"
},
{
  "start": "2024-12-04T18:10:00.000Z",
  "status": "EVENT_OVERNIGHT",
  "finished": null,
  "offtime": null
},
{
  "start": "2024-12-04T18:40:00.000Z",
  "status": "EVENT_OVERNIGHT",
  "finished": null,
  "offtime": null
},
{
  "start": "2024-12-04T19:10:00.000Z",
  "status": "EVENT_OVERNIGHT",
  "finished": null,
  "offtime": null
},
{
  "start": "2024-12-04T19:40:00.000Z",
  "status": "EVENT_OVERNIGHT",
  "finished": null,
  "offtime": null
},
{
  "start": "2024-12-04T20:10:00.000Z",
  "status": "EVENT_OVERNIGHT",
  "finished": null,
  "offtime": null
}

```



EVENTS – Field Filter examples

I want to see a simple list of runners in each race

```
{{ $baseUrl }}/events?from=04-12-2024&to=04-12-2024&fields=id,title,stageeventnumber,start,entrants[ ]&stageids=abd95869-59cc-496e-b9d6-bb48520b9fa6
```

This is for Kempton on the 4th December 2024

- NB It is recommended to submit a stageid in the event query so you can be certain about the meeting you are querying

```
id": "6af09a35-bcae-4013-8168-9e2626bb6821",
"title": "Unibet Zero% Mission Handicap",
"stageeventnumber": 8,
"start": "2024-12-04T19:40:00.000Z",
"entrants": [
  {
    "id": "49cac5dd-d7c3-4f0a-88d8-99b445d848f0",
    "name": "Alpine Stroll",
    "entrantnumber": 1,
    "stallnumber": 12,
    "status": "RUNNER"
  },
  {
    "id": "8ee130ee-8d36-4614-afee-ec8347cae7e9",
    "name": "Eton Blue",
    "entrantnumber": 2,
    "stallnumber": 5,
    "status": "RUNNER"
  },
  {
    "id": "643ace81-e7a4-4d9f-85ab-4b55871a444b",
    "name": "Silver Nightfall",
    "entrantnumber": 3,
    "stallnumber": 9,
    "status": "RUNNER"
  },
  {
    "id": "fbd27e5f-604c-491d-94b7-ed198716f308",
    "name": "Sgian Dubh",
    "entrantnumber": 4,
```




```

    "stallnumber": 6,
    "status": "RUNNER"
  },
  {
    "id": "3a45c37d-3966-494f-929b-9af650ff5e3d",
    "name": "Bohemian Breeze",
    "entrantnumber": 5,
    "stallnumber": 7,
    "status": "RUNNER"
  },
  {
    "id": "5825afa8-89e9-4394-ac2c-1106527a7306",
    "name": "Birthday Angel",
    "entrantnumber": 6,
    "stallnumber": 10,
    "status": "DOUBTFUL"
  },
  {
    "id": "07d6b012-6458-4cd1-9972-c964487326f2",
    "name": "Shamardal Star",
    "entrantnumber": 7,
    "stallnumber": 8,
    "status": "RUNNER"
  },
  {
    "id": "be54bc6e-ffc8-417c-ac60-74abadd6a3de",
    "name": "Liberated Lad",
    "entrantnumber": 8,
    "stallnumber": 3,
    "status": "RUNNER"
  },
  {
    "id": "d4246b68-e4c2-4104-9684-e81117988e42",
    "name": "Tortured Soul",
    "entrantnumber": 9,
    "stallnumber": 1,
    "status": "RUNNER"
  },
  {
    "id": "b505b52d-c6e7-43de-a5f0-f8934675401c",
    "name": "Last Empress",
    "entrantnumber": 10,
    "stallnumber": 11,
    "status": "RUNNER"
  },
  },

```



```
{
  "id": "6d1270da-0273-4a87-b256-098c6d0136a7",
  "name": "Blue Collar",
  "entrantnumber": 11,
  "stallnumber": 4,
  "status": "RUNNER"
},
{
  "id": "fa1eea5f-806c-47be-8d5c-712db1ee8f04",
  "name": "Meisterzinger",
  "entrantnumber": 12,
  "stallnumber": 2,
  "status": "RUNNER"
}
]
```

- I want to easily retrieve the silk description and retrieval URL for runners in a race

```
{{${baseUrl}}}/events?from=04-12-2024&to=04-12-2024&fields=id,title,stageeventnumber,start,entrants[ ][id,name,associations[ ][renditions{original{href,sourceText}}]]&stageids=abd95869-59cc-496e-b9d6-bb48520b9fa6
```

Partial response for brevity

```
"filters": {
  "from": "04-12-2024",
  "to": "04-12-2024",
  "fields":
    "id,title,stageeventnumber,start,entrants[ ][id,name,associations[ ][renditions{original{href,sourceText}}]]",
  "stageids": [
    "abd95869-59cc-496e-b9d6-bb48520b9fa6"
  ]
},
"items": [
  {
    "id": "dbc87146-8263-4d1a-944c-f9e85e266fcb",
    "title": "Try Unibet's New Acca Boosts Handicap",
    "stageeventnumber": 9,
    "start": "2024-12-04T20:10:00.000Z",
    "entrants": [
      {
        "id": "977e3d61-fccd-4670-9c2d-44dc9f21f537",
```



```

    "name": "Symbol of Hope",
    "associations": [
      {
        "renditions": {
          "original": {
            "href": "https://bombyx.api.press.net/v1/bombyx/silk/f6d0989c-9f03-a14e-6498-8b4a1fe8cdfe",
            "sourceText": "Yellow, dark blue diamond, chevrons on sleeves, dark blue cap, yellow diamond"
          }
        }
      }
    ],
    {
      "id": "168aae24-fb52-44e1-a5d9-9e3f8e45ffec",
      "name": "Big Bard",
      "associations": [
        {
          "renditions": {
            "original": {
              "href": "https://bombyx.api.press.net/v1/bombyx/silk/d6b426c6-3851-ad96-ecbe-4b3fb2c2d1c8",
              "sourceText": "Yellow, pink seams, yellow sleeves and cap"
            }
          }
        }
      ]
    },
    {
      "id": "b25186df-3e39-4fb2-ac4b-8728f23195e5",
      "name": "Brunel Charm",
      "associations": [
        {
          "renditions": {
            "original": {
              "href": "https://bombyx.api.press.net/v1/bombyx/silk/0fb840c3-41e0-36d5-da47-7a656cc9b39b",
              "sourceText": "Maroon, emerald green stars, sleeves and star on cap"
            }
          }
        }
      ]
    }
  ],
},

```



ENTRANTS - Fields filter examples

I want to see each betting update received by a runner in a race

{{ \$baseUrl }}/entrants/dd06959e-4090-45d8-91f7-42d12a12baf2?pagesize=25&fields=id,name,betting

Response (returns forecast price, shows and SP)

```
{
  "id": "dd06959e-4090-45d8-91f7-42d12a12baf2",
  "name": "Sand Gazelle",
  "betting": {
    "forecast": {
      "price": {
        "decimal": 6,
        "numerator": 6,
        "denominator": 1
      }
    }
  },
  "shows": [
    {
      "marketnumber": 1,
      "created": "2024-12-04T16:02:34.000Z",
      "price": {
        "decimal": 16,
        "numerator": 16,
        "denominator": 1
      }
    },
    {
      "marketnumber": 1,
      "created": "2024-12-04T16:00:33.000Z",
      "price": {
        "decimal": 14,
        "numerator": 14,
        "denominator": 1
      }
    },
    {
      "marketnumber": 1,
      "created": "2024-12-04T16:00:00.000Z",
      "price": {
        "decimal": 12,
```



```

      "numerator": 12,
      "denominator": 1
    },
    {
      "marketnumber": 1,
      "created": "2024-12-04T16:05:57.000Z",
      "price": {
        "decimal": 18,
        "numerator": 18,
        "denominator": 1
      }
    },
    {
      "marketnumber": 1,
      "created": "2024-12-04T16:06:33.000Z",
      "price": {
        "decimal": 16,
        "numerator": 16,
        "denominator": 1
      }
    },
    {
      "marketnumber": 1,
      "created": "2024-12-04T16:06:54.000Z",
      "price": {
        "decimal": 18,
        "numerator": 18,
        "denominator": 1
      }
    },
    {
      "marketnumber": 1,
      "created": "2024-12-04T16:09:46.000Z",
      "price": {
        "decimal": 20,
        "numerator": 20,
        "denominator": 1
      }
    }
  ],
  "startingprice": {
    "price": {
      "decimal": 20,
      "numerator": 20,

```

PODIUM



```
    "denominator": 1
  },
  "marketposition": null,
  "marketcount": null
}
}
```

**YOUR TRUSTED
GLOBAL SPORTS
DATA PARTNER**