API Overview

Exchange API Products

The Exchange API is for developers looking to create automated betting systems or custom betting interfaces for themselves or for Betfair customers. The Exchange API is available for the Global, Spanish and Italian Betfair Exchange

The API contains a powerful set of features that enable advanced market navigation, search, odds retrieval, bet placement and sports related data retrieval. The Exchange API is made up of the following key components:

- **Betting API** - Contains Exchange markets navigation, odds retrieval and bet placement operations. Reporting functionality relating to settled bets is also available.
- **Accounts API** - Contains account related operations such as the ability to retrieve your available account balance as well as Vendor Services API operations that are available to licensed Software Vendors
- **Exchange Stream API** - allows you to subscribe to market changes (both price and definitions) and orders.
- **Heartbeat API** - allows you to automatically cancel unmatched bets in the event of your API client/s losing connectivity.
- **Race Status API** - allows you to establish the status of a horse race or greyhound market both prior to and after the start of the race.

Other API's

- **Historical Data API** - for accessing data purchased from the Betfair Historical data service.
- **Exchange Games API** - for accessing the Exchange Games API.

Documentation

There are a number of documentation resources available:

Getting Started Guide - provides all the information required regarding licensing, login and making your first requests via the Betfair API

Reference Guide - the latest documentation for the Betfair API.

Sample Code - code samples are available in a number of programming languages.

Developer Forum - discuss your issue with our experienced developer forum community.

Demo Tools - allow you to quickly test API operations via an easy to use interface.

Translated documentation - is available in **Portuguese**, **Spanish**, & **Swedish**

Benefits & Features

The main benefits and features of the Exchange API include:

- Access to the Exchange API is free of charge for development purposes*# to all developers for personal use only.
- No data request charges for requests made via the Exchange API
- Lightweight protocol (JSON/JSON-RPC).
- Configure the depth of the best prices returned to you.
- Place LIMIT and FILL_OR_KILL orders.
- Rollup available prices - you can configure the rollup amount and type.
- Retrieve data from multiple markets in one request.
- Retrieve matched and unmatched bets and prices available via a single request.
- Search by MarketType (MATCH_ODDS, WIN, PLACE etc.) flags which remain the same, regardless of language.
- Search for in-play markets.
- View ‘result’ by selection after settlement.
- View virtual prices.

*** does not apply to commercial access**. Please see **Commercial Opportunities** for details of commercial licensing

You should use your **Delayed Application key** for development purposes.

Commercial Licencing

There are a number of different Commercial API licences available and these fit into the definitions below

Please note: We do not accept licence applications from India, Bangledesh, Sri Lanka or the UAE

Software Vendor Licence

- **We wish to create a betting app to distribute to Betfair customers.**

Please see **Vendor Program** for further information on how to apply for a Software Vendor Licence.

Odds Publisher Licence

- **We are a Betfair Affiliate & want to publish Betfair odds.**

If your already a registered Betfair Affiliate please contact our Affiliates team via Sports.Partnerships@betfair.com for more details

If your not an Affiliate, you can apply via **https://affiliates.betfair.com/ > Join Now**

Please note: Sites "under construction" will not be accepted

Betting Operator Licence

- **We are a *licensed* Betting Operator wanting to use Exchange data.**

Please contact us via https://developer.betfair.com/en/get-started/ > **Exchange API > In A Commercial Context** for further information.

Recently Updated

API Demo Tools
Apr 29, 2020 • updated by built In User • view change
Getting Started
Apr 29, 2020 • updated by built In User • view change
Betting Type Definitions
Apr 28, 2020 • updated by built In User • view change
Additional Information
Apr 15, 2020 • updated by built In User • view change
Application Keys
Apr 08, 2020 • updated by built In User • view change
placeOrders
Mar 25, 2020 • updated by built In User • view change
Exchange Stream API
Mar 20, 2020 • updated by built In User • view change
EventTypIds.xlsx
Mar 10, 2020 • attached by built In User
Soccer Selection ID.zip
Mar 10, 2020 • attached by built In User
Best Practice
Feb 27, 2020 • updated by built In User • view change
Release Notes
Feb 27, 2020 • updated by built In User • view change
API Roadmap
Feb 27, 2020 • updated by built In User • view change
Betting Exceptions
Feb 26, 2020 • updated by built In User • view change
Vendor Services API
Feb 25, 2020 • updated by built In User • view change
Javascript
Jan 24, 2020 • updated by built In User • view change

Getting Started

How Do I Get Started?

To use the Exchange API you require the following:

1. A **Betfair account**. you can open a Betfair account here
2. An **Application Key** - you can create an Application Key by following the instructions here
3. A **sessionToken -** you can create a session token by using either of the API login methods or by following the instructions here

Login

The Betfair API offers three login flows for developers, depending on the use case of your application:.

- **Non-Interactive login** - if you are building an application which will run autonomously, there is a separate login flow to follow to ensure your account remains secure.

- **Interactive login** - if you are building an application which will be used interactively, then this is the flow for you. This flow has two variants:

    - Interactive login - API method - This flow makes use of a JSON API Endpoint and is the simplest way to get started if you are looking to create your own login form.
    - Interactive login - Desktop Application- This login flow makes use of Betfair's login pages and allows your app to gracefully handle all errors and re-directions in the same way as the Betfair website

Request Headers

- All requests must include a HTTP header named **"X-Application"** containing the Application Key assigned to you.
- All requests must include a HTTP header **"X-Authentication"** containing your sessionToken.
- All requests must include a HTTP header **"Accept"** with the value application/json

Please note: The only exceptions to the above are some of the Account Operations (Vendor API) which require the **X-Authentication** HTTP header only.

You can call the API at one of two endpoints, depending on which style of request you want to use.

API Endpoints

You can make requests and place bets on UK & international markets by accessing the Global Exchange via the following endpoints.

Please find the details for the current **Betting API** endpoints:

Global Exchange

Interface	Endpoint	JSON-RPC Prefix	<method> Example
JSON-RPC	https://api.betfair.com/exchange/betting/json-rpc/v1	<method>	SportsAPING/v1.0/listMarketBook
JSON REST	https://api.betfair.com/exchange/betting/rest/v1.0/		listMarketBook/

You can make requests for your UK Exchange wallet information by accessing the Global Exchange via the following endpoints.

Please find the details for the current **Accounts API** endpoints:

Global Exchange

Interface	Endpoint	JSON-RPC Prefix	<method> Example
JSON-RPC	https://api.betfair.com/exchange/account/json-rpc/v1	<method>	AccountAPING/v1.0/getAccountFunds
JSON REST	https://api.betfair.com/exchange/account/rest/v1.0		getAccountFunds/

Spanish & Italian Exchange

Please see separate documentation for the Spanish & Italian Exchange

JSON

You can POST a request to the API at:

https://api.betfair.com/exchange/betting/rest/v1.0/<operation name>. So, to call the listEventTypes method, you would POST to: https://api.betfair.com/exchange/betting/rest/v1.0/listEventTypes/

The POST data contains the request parameters. For listEventTypes, the only required parameter is a filter to select markets. You can pass an empty filter to select all markets, in which case listEventTypes returns the EventTypes associated with all available markets.

JSON POST Data

```
{
        "filter" : { }
}
```

Python Example JSON Request

```
import requests
import json

endpoint = "https://api.betfair.com/exchange/betting/rest/v1.0/"

header = { 'X-Application' : 'APP_KEY_HERE', 'X-Authentication' : 'SESSION_TOKEN_HERE' ,'content-type' :
'application/json' }

json_req='{"filter":{ }}'

url = endpoint + "listEventTypes/"

response = requests.post(url, data=json_req, headers=header)


print json.dumps(json.loads(response.text), indent=3)
```

JSON-RPC

You can POST a request to the API using JSON-RPC at:

https://api.betfair.com/exchange/betting/json-rpc/v1

The POST data should contain a valid JSON-RPC formatted request where the "params" field contains the request parameters and the "method" field contains the API method you are calling, specified like "SportsAPING/v1.0/<operation name>.

For example, if you were calling the listCompetitions operation and passing in a filter to find all markets with a corresponding event type id of 1 (i.e., all Football markets), the POST data for the JSON-RPC endpoint would be:

Example JSON-RPC POST data

```
{
   "params": {
      "filter": {
         "eventTypeIds": [1]
      }
   },
   "jsonrpc": "2.0",
   "method": "SportsAPING/v1.0/listCompetitions",
   "id": 1
}
```

Here's a quick example Python program that uses JSON-RPC and returns the list of EventTypes (Sports) available:

JSON-RPC Python Example

```
import requests
import json

url="https://api.betfair.com/exchange/betting/json-rpc/v1"
header = { 'X-Application' : 'APP_KEY_HERE', 'X-Authentication' : 'SESSION_TOKEN' ,'content-type' : 'application
/json' }

jsonrpc_req='{"jsonrpc": "2.0", "method": "SportsAPING/v1.0/listEventTypes", "params": {"filter":{ }}, "id": 1}'

response = requests.post(url, data=jsonrpc_req, headers=header)

print json.dumps(json.loads(response.text), indent=3)
```

And the response from the above:

EventTypeResult

```
{
   "jsonrpc": "2.0",
   "result": [
      {
         "eventType": {
            "id": "468328",
            "name": "Handball"
         },
         "marketCount": 59
      },
      {
         "eventType": {
            "id": "1",
            "name": "Soccer"
         },
         "marketCount": 14792
      },
      {
         "eventType": {
            "id": "2",
            "name": "Tennis"
         },
         "marketCount": 51
      },
      {
         "eventType": {
            "id": "3",
            "name": "Golf"
         },
         "marketCount": 12
      },
      {
         "eventType": {
            "id": "4",
            "name": "Cricket"
         },
         "marketCount": 139
      },
      {
         "eventType": {
            "id": "5",
            "name": "Rugby Union"
         },
         "marketCount": 100
      },
      {
         "eventType": {
            "id": "6",
            "name": "Boxing"
         },
         "marketCount": 12
      },
      {
         "eventType": {
            "id": "7",
            "name": "Horse Racing"
         },
         "marketCount": 187
      },
      {
         "eventType": {
            "id": "8",
            "name": "Motor Sport"
         },
         "marketCount": 3
      },
      {
         "eventType": {
            "id": "7524",
            "name": "Ice Hockey"
         },
         "marketCount": 8
      },
      {
         "eventType": {
            "id": "10",
            "name": "Special Bets"
         },
         "marketCount": 30
      },
      {
         "eventType": {
            "id": "451485",
            "name": "Winter Sports"
         },
         "marketCount": 7
      },
      {
         "eventType": {
            "id": "7522",
            "name": "Basketball"
         },
         "marketCount": 559
      },
      {
         "eventType": {
            "id": "1477",
            "name": "Rugby League"
         },
         "marketCount": 3
      },
      {
         "eventType": {
            "id": "4339",
            "name": "Greyhound Racing"
         },
         "marketCount": 269
      },
      {
         "eventType": {
            "id": "2378961",
            "name": "Politics"
         },
         "marketCount": 19
      },
      {
         "eventType": {
            "id": "6231",
            "name": "Financial Bets"
         },
         "marketCount": 51
      },
      {
         "eventType": {
            "id": "998917",
            "name": "Volleyball"
         },
         "marketCount": 69
      },
      {
         "eventType": {
            "id": "998919",
            "name": "Bandy"
         },
         "marketCount": 2
      },
      {
         "eventType": {
            "id": "998918",
            "name": "Bowls"
         },
         "marketCount": 10
      },
      {
         "eventType": {
            "id": "3503",
            "name": "Darts"
         },
         "marketCount": 446
      },
      {
         "eventType": {
            "id": "72382",
            "name": "Pool"
         },
         "marketCount": 1
      },
      {
         "eventType": {
            "id": "6422",
            "name": "Snooker"
         },
         "marketCount": 3
      },
      {
         "eventType": {
            "id": "6423",
            "name": "American Football"
         },
         "marketCount": 86
      },
      {
         "eventType": {
            "id": "7511",
            "name": "Baseball"
         },
         "marketCount": 1
      }
   ],
   "id": 1
}
```

API Demo Tools

Our **Demo Tools** can be used by developers for quick experimentation and interaction with the production API system

Example Requests

This section shows how you might call the Betting API to retrieve information.

This section includes examples of how to request the following information in jsonrpc format:

- lRequest A List Of Available Event Types (**listEventTypes**)
- Request a List of Events for an Event Type (**listEvents**)
- Request the Market Information for an Event (**listMarketCatalogue**)
- Horse Racing - Today's Win & Place Markets
- Request a List of Football Competitions (**listCompetitions**)
- Request Market Prices (**listMarketBook**)
- Placing a Bet **(placeOrders)**
- Placing a Betfair SP Bet - MARKET_ON_CLOSE **(placeOrders)**
- Placing a Betfair SP Bet - LIMIT_ON_CLOSE **(placeOrders)**
- Retrieving Details of Bet/s Placed on a Market/s **(listCurrentOrders)**
- Retrieving the Result of a Settled **(listMarketBook)**
- Retrieving Details of Bets on a Settled Market - including P&L & Commission paid **(listClearedOrders)**

Request A List Of Available Event Types

You can make a request using the listEventTypes service which will return a response containing the eventTypes (e.g. Soccer, Horse Racing etc.) that are currently available on Betfair.

listEventTypes Request

```
[
    {
        "jsonrpc": "2.0",
        "method": "SportsAPING/v1.0/listEventTypes",
        "params": {
            "filter": {}
        },
        "id": 1
    }
]
```

listEventTypes Response

```
[
    {
        "jsonrpc": "2.0",
        "result": [
            {
                "eventType": {
                    "id": "468328",
                    "name": "Handball"
                },
                "marketCount": 11
            },
            {
                "eventType": {
                    "id": "1",
                    "name": "Soccer"
                },
                "marketCount": 25388
            },
            {
                "eventType": {
                    "id": "2",
                    "name": "Tennis"
                },
                "marketCount": 402
            },
            {
                "eventType": {
                    "id": "3",
                    "name": "Golf"
                },
                "marketCount": 79
            },
            {
                "eventType": {
                    "id": "4",
                    "name": "Cricket"
                },
                "marketCount": 192
            },
            {
                "eventType": {
                    "id": "5",
                    "name": "Rugby Union"
                },
                "marketCount": 233
            },
            {
                "eventType": {
                    "id": "6",
                    "name": "Boxing"
                },
                "marketCount": 18
            },
            {
                "eventType": {
                    "id": "7",
                    "name": "Horse Racing"
                },
                "marketCount": 398
            },
            {
                "eventType": {
                    "id": "8",
                    "name": "Motor Sport"
                },
                "marketCount": 50
            },
            {
                "eventType": {
                    "id": "7524",
                    "name": "Ice Hockey"
                },
                "marketCount": 521
            },
            {
                "eventType": {
                    "id": "10",
                    "name": "Special Bets"
                },
                "marketCount": 39
            },
            {
                "eventType": {
                    "id": "451485",
                    "name": "Winter Sports"
                },
                "marketCount": 7
            },
            {
                "eventType": {
                    "id": "11",
                    "name": "Cycling"
                },
                "marketCount": 1
            },
            {
                "eventType": {
                    "id": "136332",
                    "name": "Chess"
                },
                "marketCount": 1
            },
            {
                "eventType": {
                    "id": "7522",
                    "name": "Basketball"
                },
                "marketCount": 617
            },
            {
                "eventType": {
                    "id": "1477",
                    "name": "Rugby League"
                },
                "marketCount": 91
            },
            {
                "eventType": {
                    "id": "4339",
                    "name": "Greyhound Racing"
                },
                "marketCount": 298
            },
            {
                "eventType": {
                    "id": "6231",
                    "name": "Financial Bets"
                },
                "marketCount": 44
            },
            {
                "eventType": {
                    "id": "2378961",
                    "name": "Politics"
                },
                "marketCount": 23
            },
            {
                "eventType": {
                    "id": "998917",
                    "name": "Volleyball"
                },
                "marketCount": 66
            },
            {
                "eventType": {
                    "id": "998919",
                    "name": "Bandy"
                },
                "marketCount": 4
            },
            {
                "eventType": {
                    "id": "998918",
                    "name": "Bowls"
                },
                "marketCount": 17
            },
            {
                "eventType": {
                    "id": "26420387",
                    "name": "Mixed Martial Arts"
                },
                "marketCount": 52
            },
            {
                "eventType": {
                    "id": "3503",
                    "name": "Darts"
                },
                "marketCount": 21
            },
            {
                "eventType": {
                    "id": "2152880",
                    "name": "Gaelic Games"
                },
                "marketCount": 2
            },
            {
                "eventType": {
                    "id": "6422",
                    "name": "Snooker"
                },
                "marketCount": 22
            },
            {
                "eventType": {
                    "id": "6423",
                    "name": "American Football"
                },
                "marketCount": 171
            },
            {
                "eventType": {
                    "id": "315220",
                    "name": "Poker"
                },
                "marketCount": 2
            },
            {
                "eventType": {
                    "id": "7511",
                    "name": "Baseball"
                },
                "marketCount": 7
            }
        ],
        "id": 1
    }
]
```

Request a List of Events for an Event Type

The below example demonstrates how to retrieve a list of events using listEvents (eventIds) for a specific event type. The request shows how to retrieve all Soccer events that are taking place in a single day.

listEvents Request

```
[
    {
        "jsonrpc": "2.0",
        "method": "SportsAPING/v1.0/listEvents",
        "params": {
            "filter": {
                "eventTypeIds": [
                    "1"
                ],
                "marketStartTime": {
                    "from": "2014-03-13T00:00:00Z",
                    "to": "2014-03-13T23:59:00Z"
                }
            }
        },
        "id": 1
    }
]
```

listEvents Response

```
[
    {
        "jsonrpc": "2.0",
        "result": [
            {
                "event": {
                    "id": "27165668",
                    "name": "Al-Wahda (KSA) v Hajer (KSA)",
                    "countryCode": "SA",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T13:30:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27165665",
                    "name": "Al Hussein v Mansheyat Bani Hasan",
                    "countryCode": "JO",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T15:00:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27165425",
                    "name": "Daily Goals",
                    "countryCode": "GB",
                    "timezone": "Europe/London",
                    "openDate": "2014-03-13T18:00:00.000Z"
                },
                "marketCount": 1
            },
            {
                "event": {
                    "id": "27165667",
                    "name": "Al Jeel v Al Draih",
                    "countryCode": "SA",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T12:45:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27165677",
                    "name": "Daventry Town v Kettering",
                    "countryCode": "GB",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T19:45:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27160160",
                    "name": "Porto v Napoli",
                    "countryCode": "PT",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T18:00:00.000Z"
                },
                "marketCount": 84
            },
            {
                "event": {
                    "id": "27162435",
                    "name": "Bishops Stortford v Hayes And Yeading",
                    "countryCode": "GB",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T19:45:00.000Z"
                },
                "marketCount": 2
            },
            {
                "event": {
                    "id": "27166333",
                    "name": "Bosnia U19 v Serbia U19",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T12:30:00.000Z"
                },
                "marketCount": 25
            },
            {
                "event": {
                    "id": "27162436",
                    "name": "Maidenhead v Gosport Borough",
                    "countryCode": "GB",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T19:45:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27165673",
                    "name": "ASA Tel Aviv Uni (W) v FC Ramat Hasharon (W)",
                    "countryCode": "IL",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T17:15:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27164435",
                    "name": "Forest Green v Braintree",
                    "countryCode": "GB",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T19:45:00.000Z"
                },
                "marketCount": 15
            },
            {
                "event": {
                    "id": "27165685",
                    "name": "FC Samtredia v FC Betlemi Keda",
                    "countryCode": "GE",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T11:00:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27165684",
                    "name": "FC Lokomotivi Tbilisi v FC Saburtalo Tbilisi",
                    "countryCode": "GE",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T11:00:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27165686",
                    "name": "FC Sasco Tbilisi v Matchak Khelvachauri",
                    "countryCode": "GE",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T11:00:00.000Z"
                },
                "marketCount": 18
            },
            {
                "event": {
                    "id": "27165680",
                    "name": "FAR Rabat v Maghreb Fes",
                    "countryCode": "MA",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T15:30:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27165683",
                    "name": "FC Kolkheti Khobi v Samgurali Tskaltubo",
                    "countryCode": "GE",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T11:00:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27165682",
                    "name": "FC Dila Gori II v FC Dinamo Batumi",
                    "countryCode": "GE",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T11:00:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27165693",
                    "name": "Tilbury FC v Redbridge",
                    "countryCode": "GB",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T19:45:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27165688",
                    "name": "HUJK Emmaste v Kohtla-Jarve JK Jarve",
                    "countryCode": "EE",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T17:00:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27165690",
                    "name": "M Kishronot Hadera (W) v Maccabi Holon FC (W)",
                    "countryCode": "IL",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T17:30:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27166225",
                    "name": "Litex Lovech v Cherno More",
                    "countryCode": "BG",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T15:30:00.000Z"
                },
                "marketCount": 27
            },
            {
                "event": {
                    "id": "27162412",
                    "name": "KR Reykjavik v IA Akranes",
                    "countryCode": "IS",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T19:00:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27162473",
                    "name": "Atletico Huila v Tolima",
                    "countryCode": "CO",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T23:00:00.000Z"
                },
                "marketCount": 27
            },
            {
                "event": {
                    "id": "27162413",
                    "name": "KV v Selfoss",
                    "countryCode": "IS",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T21:00:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27165159",
                    "name": "August Town FC v Boys Town FC",
                    "countryCode": "JM",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T20:30:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27165161",
                    "name": "Bogota v CD Barranquilla",
        "countryCode": "CO",
        "timezone": "GMT",
        "openDate": "2014-03-13T20:00:00.000Z"
    },
    "marketCount": 20
},
{
    "event": {
        "id": "27166474",
        "name": "Brasilia FC v Formosa",
        "countryCode": "BR",
        "timezone": "GMT",
        "openDate": "2014-03-13T19:00:00.000Z"
    },
    "marketCount": 15
},
{
    "event": {
        "id": "27162538",
        "name": "Arsenal FC v Penarol",
        "countryCode": "AR",
        "timezone": "GMT",
        "openDate": "2014-03-13T22:00:00.000Z"
    },
    "marketCount": 40
},
{
    "event": {
        "id": "27166478",
        "name": "Ware FC v AFC Sudbury",
        "countryCode": "GB",
        "timezone": "GMT",
        "openDate": "2014-03-13T19:45:00.000Z"
    },
    "marketCount": 15
},
{
    "event": {
        "id": "27165505",
        "name": "Tomsk v Tyumen",
        "countryCode": "RU",
        "timezone": "GMT",
        "openDate": "2014-03-13T11:30:00.000Z"
    },
    "marketCount": 28
},
{
    "event": {
        "id": "27166477",
        "name": "Needham Market FC v Thurrock",
        "countryCode": "GB",
        "timezone": "GMT",
        "openDate": "2014-03-13T19:45:00.000Z"
    },
    "marketCount": 15
},
{
    "event": {
        "id": "27160154",
        "name": "Lyon v Plzen",
        "countryCode": "FR",
        "timezone": "GMT",
        "openDate": "2014-03-13T20:05:00.000Z"
    },
    "marketCount": 41
},
{
    "event": {
        "id": "27160155",
        "name": "Ludogorets v Valencia",
        "countryCode": "BG",
                "timezone": "GMT",
                "openDate": "2014-03-13T18:00:00.000Z"
            },
            "marketCount": 84
        },
        {
            "event": {
                "id": "27160152",
                "name": "Tottenham v Benfica",
                "countryCode": "GB",
                "timezone": "GMT",
                "openDate": "2014-03-13T20:05:00.000Z"
            },
            "marketCount": 84
        },
        {
            "event": {
                "id": "27162428",
                "name": "Wadi Degla v El Shorta",
                "countryCode": "EG",
                "timezone": "GMT",
                "openDate": "2014-03-13T13:00:00.000Z"
            },
            "marketCount": 20
        },
        {
            "event": {
                "id": "27160158",
                "name": "FC Basel v Red Bull Salzburg",
                "countryCode": "CH",
                "timezone": "GMT",
                "openDate": "2014-03-13T18:00:00.000Z"
            },
            "marketCount": 84
        },
        {
            "event": {
                "id": "27162427",
                "name": "Ismaily v El Qanah",
                "countryCode": "EG",
                "timezone": "GMT",
                "openDate": "2014-03-13T13:00:00.000Z"
            },
            "marketCount": 20
        },
        {
            "event": {
                "id": "27160159",
                "name": "AZ Alkmaar v Anzhi Makhachkala",
                "countryCode": "NL",
                "timezone": "GMT",
                "openDate": "2014-03-13T20:05:00.000Z"
            },
            "marketCount": 41
        },
        {
            "event": {
                "id": "27162426",
                "name": "Al Ahly v El Entag El Harby",
                "countryCode": "EG",
                "timezone": "GMT",
                "openDate": "2014-03-13T15:30:00.000Z"
            },
            "marketCount": 15
        },
        {
            "event": {
                "id": "27160156",
                "name": "Sevilla v Betis",
                "countryCode": "ES",
                "timezone": "GMT",
                    "openDate": "2014-03-13T20:05:00.000Z"
                },
                "marketCount": 84
            },
            {
                "event": {
                    "id": "27160157",
                    "name": "Juventus v Fiorentina",
                    "countryCode": "IT",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T20:05:00.000Z"
                },
                "marketCount": 84
            },
            {
                "event": {
                    "id": "27166336",
                    "name": "Becamex Binh Duong U19 v Khanh Hoa U19",
                    "countryCode": "VN",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T11:00:00.000Z"
                },
                "marketCount": 15
            },
            {
                "event": {
                    "id": "27163800",
                    "name": "Lokomotiv Sofia v Chernomorets Burgas",
                    "countryCode": "BG",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T12:00:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27162481",
                    "name": "Ljungskile v Torslanda",
                    "countryCode": "SE",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T18:00:00.000Z"
                },
                "marketCount": 27
            },
            {
                "event": {
                    "id": "27166338",
                    "name": "H Ironi Petah Tikva (W) v Maccabi Beer Sheva (W)",
                    "countryCode": "IL",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T18:15:00.000Z"
                },
                "marketCount": 15
            },
            {
                "event": {
                    "id": "27163801",
                    "name": "Concord Rangers v Havant and W",
                    "countryCode": "GB",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T19:45:00.000Z"
                },
                "marketCount": 2
            },
            {
                "event": {
                    "id": "27166340",
                    "name": "Maccabi Ironi Bat Yam v Hapoel Mahane Yehuda",
                    "countryCode": "IL",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T17:00:00.000Z"
                },
                "marketCount": 15
            },
            {
                "event": {
                    "id": "27162418",
                    "name": "Courts Young Lions v Woodlands Wellington",
                    "countryCode": "SG",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T11:30:00.000Z"
                },
                "marketCount": 20
            },
            {
                "event": {
                    "id": "27162417",
                    "name": "Balestier Khalsa v Tanjong Pagar Utd",
                    "countryCode": "SG",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T11:30:00.000Z"
                },
                "marketCount": 20
            }
        ],
        "id": 1
    }
]
```

Request the Market Information for an Event

The below example demonstrates how to retrieve all the market information that belongs to an event (excluding price data) using listMarketCatalogue. You can include one or more eventId's in the requests provide that you stay within the Market Data Limits

listMarketCatalogue Request

```
[
    {
        "jsonrpc": "2.0",
        "method": "SportsAPING/v1.0/listMarketCatalogue",
        "params": {
            "filter": {
                "eventIds": [
                    "27165685"
                ]
            },
            "maxResults": "200",
            "marketProjection": [
                "COMPETITION",
                "EVENT",
                "EVENT_TYPE",
                "RUNNER_DESCRIPTION",
                "RUNNER_METADATA",
                "MARKET_START_TIME"
            ]
        },
        "id": 1
    }
]
```

listMarketCatalogue Response

```
[
    {
        "jsonrpc": "2.0",
        "result": [
            {
                "marketId": "1.113197547",
                "marketName": "FC Betlemi Keda +1",
                "marketStartTime": "2014-03-13T11:00:00.000Z",
                "totalMatched": 12,
                "runners": [
                    {
                        "selectionId": 6843871,
                        "runnerName": "FC Betlemi Keda +1",
                        "handicap": 0,
                        "sortPriority": 1,
                        "metadata": {
                            "runnerId": "63123618"
                        }
                    },
                    {
                        "selectionId": 6830600,
                        "runnerName": "FC Samtredia -1",
                        "handicap": 0,
                        "sortPriority": 2,
                        "metadata": {
                            "runnerId": "63123619"
                        }
                    },
                    {
                        "selectionId": 151478,
                        "runnerName": "Draw",
                        "handicap": 0,
                        "sortPriority": 3,
                        "metadata": {
                            "runnerId": "63123620"
                        }
                    }
                ],
                "eventType": {
                    "id": "1",
                    "name": "Soccer"
                },
                "competition": {
                    "id": "2356065",
                    "name": "Pirveli Liga"
                },
                "event": {
                    "id": "27165685",
                    "name": "FC Samtredia v FC Betlemi Keda",
                    "countryCode": "GE",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T11:00:00.000Z"
                }
            },
            {
                "marketId": "1.113197546",
                "marketName": "FC Samtredia +1",
                "marketStartTime": "2014-03-13T11:00:00.000Z",
                "totalMatched": 0,
                "runners": [
                    {
                        "selectionId": 6830597,
                        "runnerName": "FC Samtredia +1",
                        "handicap": 0,
                        "sortPriority": 1,
                        "metadata": {
                            "runnerId": "63123615"
                        }
                    },
                    {
                        "selectionId": 6843874,
                        "runnerName": "FC Betlemi Keda -1",
                        "handicap": 0,
                        "sortPriority": 2,
                        "metadata": {
                            "runnerId": "63123616"
                        }
                    },
                    {
                        "selectionId": 151478,
                        "runnerName": "Draw",
                        "handicap": 0,
                        "sortPriority": 3,
                        "metadata": {
                            "runnerId": "63123617"
                        }
                    }
                ],
                "eventType": {
                    "id": "1",
                    "name": "Soccer"
                },
                "competition": {
                    "id": "2356065",
                    "name": "Pirveli Liga"
                },
                "event": {
                    "id": "27165685",
                    "name": "FC Samtredia v FC Betlemi Keda",
                    "countryCode": "GE",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T11:00:00.000Z"
                }
            },
            {
                "marketId": "1.113197492",
                "marketName": "Total Goals",
                "marketStartTime": "2014-03-13T11:00:00.000Z",
                "totalMatched": 246.82,
                "runners": [
                    {
                        "selectionId": 285469,
                        "runnerName": "1 goals or more",
                        "handicap": -1,
                        "sortPriority": 1,
                        "metadata": {
                            "runnerId": "63123486"
                        }
                    },
                    {
                        "selectionId": 285470,
                        "runnerName": "2 goals or more",
                        "handicap": -2,
                        "sortPriority": 2,
                        "metadata": {
                            "runnerId": "63123487"
                        }
                    },
                    {
                        "selectionId": 285471,
                        "runnerName": "3 goals or more",
                        "handicap": -3,
                        "sortPriority": 3,
                        "metadata": {
                            "runnerId": "63123488"
                        }
                    },
                    {
                        "selectionId": 2795170,
                        "runnerName": "4 goals or more",
                        "handicap": -4,
                        "sortPriority": 4,
                        "metadata": {
                            "runnerId": "63123489"
                        }
                    },
                    {
```

```
                        "selectionId": 285473,
                        "runnerName": "5 goals or more",
                        "handicap": -5,
                        "sortPriority": 5,
                        "metadata": {
                            "runnerId": "63123490"
                        }
                    },
                    {
                        "selectionId": 285474,
                        "runnerName": "6 goals or more",
                        "handicap": -6,
                        "sortPriority": 6,
                        "metadata": {
                            "runnerId": "63123491"
                        }
                    },
                    {
                        "selectionId": 8215951,
                        "runnerName": "7 goals or more",
                        "handicap": -7,
                        "sortPriority": 7,
                        "metadata": {
                            "runnerId": "63123492"
                        }
                    }
                ],
                "eventType": {
                    "id": "1",
                    "name": "Soccer"
                },
                "competition": {
                    "id": "2356065",
                    "name": "Pirveli Liga"
                },
                "event": {
                    "id": "27165685",
                    "name": "FC Samtredia v FC Betlemi Keda",
                    "countryCode": "GE",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T11:00:00.000Z"
                }
            },
            {
                "marketId": "1.113197491",
                "marketName": "Match Odds",
                "marketStartTime": "2014-03-13T11:00:00.000Z",
                "totalMatched": 7707.52,
                "runners": [
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": 0,
                        "sortPriority": 1,
                        "metadata": {
                            "runnerId": "63123483"
                        }
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": 0,
                        "sortPriority": 2,
                        "metadata": {
                            "runnerId": "63123484"
                        }
                    },
                    {
                        "selectionId": 58805,
                        "runnerName": "The Draw",
                        "handicap": 0,
                    "sortPriority": 3,
                    "metadata": {
                        "runnerId": "63123485"
                    }
                }
            ],
            "eventType": {
                "id": "1",
                "name": "Soccer"
            },
            "competition": {
                "id": "2356065",
                "name": "Pirveli Liga"
            },
            "event": {
                "id": "27165685",
                "name": "FC Samtredia v FC Betlemi Keda",
                "countryCode": "GE",
                "timezone": "GMT",
                "openDate": "2014-03-13T11:00:00.000Z"
            }
        },
        {
            "marketId": "1.113197550",
            "marketName": "Both teams to Score?",
            "marketStartTime": "2014-03-13T11:00:00.000Z",
            "totalMatched": 14.78,
            "runners": [
                {
                    "selectionId": 30246,
                    "runnerName": "Yes",
                    "handicap": 0,
                    "sortPriority": 1,
                    "metadata": {
                        "runnerId": "63123625"
                    }
                },
                {
                    "selectionId": 30247,
                    "runnerName": "No",
                    "handicap": 0,
                    "sortPriority": 2,
                    "metadata": {
                        "runnerId": "63123626"
                    }
                }
            ],
            "eventType": {
                "id": "1",
                "name": "Soccer"
            },
            "competition": {
                "id": "2356065",
                "name": "Pirveli Liga"
            },
            "event": {
                "id": "27165685",
                "name": "FC Samtredia v FC Betlemi Keda",
                "countryCode": "GE",
                "timezone": "GMT",
                "openDate": "2014-03-13T11:00:00.000Z"
            }
        },
        {
            "marketId": "1.113197501",
            "marketName": "Next Goal",
            "marketStartTime": "2014-03-13T11:00:00.000Z",
            "totalMatched": 3.34,
            "runners": [
                {
                    "selectionId": 6830593,
                    "runnerName": "FC Samtredia",
                    "handicap": 0,
                    "sortPriority": 1,
                    "metadata": {
                        "runnerId": "63123495"
                    }
                },
                {
                    "selectionId": 6843866,
                    "runnerName": "FC Betlemi Keda",
                    "handicap": 0,
                    "sortPriority": 2,
                    "metadata": {
                        "runnerId": "63123496"
                    }
                },
                {
                    "selectionId": 69852,
                    "runnerName": "No Goal",
                    "handicap": 0,
                    "sortPriority": 3,
                    "metadata": {
                        "runnerId": "63123497"
                    }
                }
            ],
            "eventType": {
                "id": "1",
                "name": "Soccer"
            },
            "competition": {
                "id": "2356065",
                "name": "Pirveli Liga"
            },
            "event": {
                "id": "27165685",
                "name": "FC Samtredia v FC Betlemi Keda",
                "countryCode": "GE",
                "timezone": "GMT",
                "openDate": "2014-03-13T11:00:00.000Z"
            }
        },
        {
            "marketId": "1.113197502",
            "marketName": "Over/Under 6.5 Goals",
            "marketStartTime": "2014-03-13T11:00:00.000Z",
            "totalMatched": 1255.79,
            "runners": [
                {
                    "selectionId": 2542448,
                    "runnerName": "Under 6.5 Goals",
                    "handicap": 0,
                    "sortPriority": 1,
                    "metadata": {
                        "runnerId": "63123498"
                    }
                },
                {
                    "selectionId": 2542449,
                    "runnerName": "Over 6.5 Goals",
                    "handicap": 0,
                    "sortPriority": 2,
                    "metadata": {
                        "runnerId": "63123499"
                    }
                }
            ],
            "eventType": {
                "id": "1",
                "name": "Soccer"
            },
                "competition": {
                    "id": "2356065",
                    "name": "Pirveli Liga"
                },
                "event": {
                    "id": "27165685",
                    "name": "FC Samtredia v FC Betlemi Keda",
                    "countryCode": "GE",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T11:00:00.000Z"
                }
            },
            {
                "marketId": "1.113197505",
                "marketName": "Correct Score",
                "marketStartTime": "2014-03-13T11:00:00.000Z",
                "totalMatched": 2380.92,
                "runners": [
                    {
                        "selectionId": 1,
                        "runnerName": "0 - 0",
                        "handicap": 0,
                        "sortPriority": 1,
                        "metadata": {
                            "runnerId": "63123505"
                        }
                    },
                    {
                        "selectionId": 4,
                        "runnerName": "0 - 1",
                        "handicap": 0,
                        "sortPriority": 2,
                        "metadata": {
                            "runnerId": "63123506"
                        }
                    },
                    {
                        "selectionId": 9,
                        "runnerName": "0 - 2",
                        "handicap": 0,
                        "sortPriority": 3,
                        "metadata": {
                            "runnerId": "63123507"
                        }
                    },
                    {
                        "selectionId": 16,
                        "runnerName": "0 - 3",
                        "handicap": 0,
                        "sortPriority": 4,
                        "metadata": {
                            "runnerId": "63123508"
                        }
                    },
                    {
                        "selectionId": 2,
                        "runnerName": "1 - 0",
                        "handicap": 0,
                        "sortPriority": 5,
                        "metadata": {
                            "runnerId": "63123509"
                        }
                    },
                    {
                        "selectionId": 3,
                        "runnerName": "1 - 1",
                        "handicap": 0,
                        "sortPriority": 6,
                        "metadata": {
                            "runnerId": "63123510"
                        }
                    },
                    {
                        "selectionId": 8,
                        "runnerName": "1 - 2",
                        "handicap": 0,
                        "sortPriority": 7,
                        "metadata": {
                            "runnerId": "63123511"
                        }
                    },
                    {
                        "selectionId": 15,
                        "runnerName": "1 - 3",
                        "handicap": 0,
                        "sortPriority": 8,
                        "metadata": {
                            "runnerId": "63123512"
                        }
                    },
                    {
                        "selectionId": 5,
                        "runnerName": "2 - 0",
                        "handicap": 0,
                        "sortPriority": 9,
                        "metadata": {
                            "runnerId": "63123513"
                        }
                    },
                    {
                        "selectionId": 6,
                        "runnerName": "2 - 1",
                        "handicap": 0,
                        "sortPriority": 10,
                        "metadata": {
                            "runnerId": "63123514"
                        }
                    },
                    {
                        "selectionId": 7,
                        "runnerName": "2 - 2",
                        "handicap": 0,
                        "sortPriority": 11,
                        "metadata": {
                            "runnerId": "63123515"
                        }
                    },
                    {
                        "selectionId": 14,
                        "runnerName": "2 - 3",
                        "handicap": 0,
                        "sortPriority": 12,
                        "metadata": {
                            "runnerId": "63123516"
                        }
                    },
                    {
                        "selectionId": 10,
                        "runnerName": "3 - 0",
                        "handicap": 0,
                        "sortPriority": 13,
                        "metadata": {
                            "runnerId": "63123517"
                        }
                    },
                    {
                        "selectionId": 11,
                        "runnerName": "3 - 1",
                        "handicap": 0,
                        "sortPriority": 14,
                        "metadata": {
                            "runnerId": "63123518"
                    }
                },
                {
                    "selectionId": 12,
                    "runnerName": "3 - 2",
                    "handicap": 0,
                    "sortPriority": 15,
                    "metadata": {
                        "runnerId": "63123519"
                    }
                },
                {
                    "selectionId": 13,
                    "runnerName": "3 - 3",
                    "handicap": 0,
                    "sortPriority": 16,
                    "metadata": {
                        "runnerId": "63123520"
                    }
                },
                {
                    "selectionId": 4506345,
                    "runnerName": "Any Unquoted ",
                    "handicap": 0,
                    "sortPriority": 17,
                    "metadata": {
                        "runnerId": "63123521"
                    }
                }
            ],
            "eventType": {
                "id": "1",
                "name": "Soccer"
            },
            "competition": {
                "id": "2356065",
                "name": "Pirveli Liga"
            },
            "event": {
                "id": "27165685",
                "name": "FC Samtredia v FC Betlemi Keda",
                "countryCode": "GE",
                "timezone": "GMT",
                "openDate": "2014-03-13T11:00:00.000Z"
            }
        },
        {
            "marketId": "1.113197506",
            "marketName": "Over/Under 2.5 Goals",
            "marketStartTime": "2014-03-13T11:00:00.000Z",
            "totalMatched": 4149.36,
            "runners": [
                {
                    "selectionId": 47972,
                    "runnerName": "Under 2.5 Goals",
                    "handicap": 0,
                    "sortPriority": 1,
                    "metadata": {
                        "runnerId": "63123522"
                    }
                },
                {
                    "selectionId": 47973,
                    "runnerName": "Over 2.5 Goals",
                    "handicap": 0,
                    "sortPriority": 2,
                    "metadata": {
                        "runnerId": "63123523"
                    }
                }
            ],
            "eventType": {
                "id": "1",
                "name": "Soccer"
            },
            "competition": {
                "id": "2356065",
                "name": "Pirveli Liga"
            },
            "event": {
                "id": "27165685",
                "name": "FC Samtredia v FC Betlemi Keda",
                "countryCode": "GE",
                "timezone": "GMT",
                "openDate": "2014-03-13T11:00:00.000Z"
            }
        },
        {
            "marketId": "1.113197504",
            "marketName": "Over/Under 5.5 Goals",
            "marketStartTime": "2014-03-13T11:00:00.000Z",
            "totalMatched": 2216.24,
            "runners": [
                {
                    "selectionId": 1485567,
                    "runnerName": "Under 5.5 Goals",
                    "handicap": 0,
                    "sortPriority": 1,
                    "metadata": {
                        "runnerId": "63123503"
                    }
                },
                {
                    "selectionId": 1485568,
                    "runnerName": "Over 5.5 Goals",
                    "handicap": 0,
                    "sortPriority": 2,
                    "metadata": {
                        "runnerId": "63123504"
                    }
                }
            ],
            "eventType": {
                "id": "1",
                "name": "Soccer"
            },
            "competition": {
                "id": "2356065",
                "name": "Pirveli Liga"
            },
            "event": {
                "id": "27165685",
                "name": "FC Samtredia v FC Betlemi Keda",
                "countryCode": "GE",
                "timezone": "GMT",
                "openDate": "2014-03-13T11:00:00.000Z"
            }
        },
        {
            "marketId": "1.113197509",
            "marketName": "Half Time/Full Time",
            "marketStartTime": "2014-03-13T11:00:00.000Z",
            "totalMatched": 97.3,
            "runners": [
                {
                    "selectionId": 6830596,
                    "runnerName": "FC Samtredia/FC Samtredia",
                    "handicap": 0,
                    "sortPriority": 1,
                    "metadata": {
                        "runnerId": "63123536"
                    }
                    },
                    {
                        "selectionId": 6830599,
                        "runnerName": "FC Samtredia/Draw",
                        "handicap": 0,
                        "sortPriority": 2,
                        "metadata": {
                            "runnerId": "63123537"
                        }
                    },
                    {
                        "selectionId": 8380726,
                        "runnerName": "FC Samtredia/FC Betlemi Ked",
                        "handicap": 0,
                        "sortPriority": 3,
                        "metadata": {
                            "runnerId": "63123538"
                        }
                    },
                    {
                        "selectionId": 6830595,
                        "runnerName": "Draw/FC Samtredia",
                        "handicap": 0,
                        "sortPriority": 4,
                        "metadata": {
                            "runnerId": "63123539"
                        }
                    },
                    {
                        "selectionId": 3710152,
                        "runnerName": "Draw/Draw",
                        "handicap": 0,
                        "sortPriority": 5,
                        "metadata": {
                            "runnerId": "63123540"
                        }
                    },
                    {
                        "selectionId": 6843869,
                        "runnerName": "Draw/FC Betlemi Ked",
                        "handicap": 0,
                        "sortPriority": 6,
                        "metadata": {
                            "runnerId": "63123541"
                        }
                    },
                    {
                        "selectionId": 8380727,
                        "runnerName": "FC Betlemi Ked/FC Samtredia",
                        "handicap": 0,
                        "sortPriority": 7,
                        "metadata": {
                            "runnerId": "63123542"
                        }
                    },
                    {
                        "selectionId": 6843873,
                        "runnerName": "FC Betlemi Ked/Draw",
                        "handicap": 0,
                        "sortPriority": 8,
                        "metadata": {
                            "runnerId": "63123543"
                        }
                    },
                    {
                        "selectionId": 6843870,
                        "runnerName": "FC Betlemi Ked/FC Betlemi Ked",
                        "handicap": 0,
                        "sortPriority": 9,
                        "metadata": {
                            "runnerId": "63123544"
                        }
                    }
                ],
                "eventType": {
                    "id": "1",
                    "name": "Soccer"
                },
                "competition": {
                    "id": "2356065",
                    "name": "Pirveli Liga"
                },
                "event": {
                    "id": "27165685",
                    "name": "FC Samtredia v FC Betlemi Keda",
                    "countryCode": "GE",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T11:00:00.000Z"
                }
            },
            {
                "marketId": "1.113197510",
                "marketName": "Over/Under 1.5 Goals",
                "marketStartTime": "2014-03-13T11:00:00.000Z",
                "totalMatched": 1879.69,
                "runners": [
                    {
                        "selectionId": 1221385,
                        "runnerName": "Under 1.5 Goals",
                        "handicap": 0,
                        "sortPriority": 1,
                        "metadata": {
                            "runnerId": "63123545"
                        }
                    },
                    {
                        "selectionId": 1221386,
                        "runnerName": "Over 1.5 Goals",
                        "handicap": 0,
                        "sortPriority": 2,
                        "metadata": {
                            "runnerId": "63123546"
                        }
                    }
                ],
                "eventType": {
                    "id": "1",
                    "name": "Soccer"
                },
                "competition": {
                    "id": "2356065",
                    "name": "Pirveli Liga"
                },
                "event": {
                    "id": "27165685",
                    "name": "FC Samtredia v FC Betlemi Keda",
                    "countryCode": "GE",
                    "timezone": "GMT",
                    "openDate": "2014-03-13T11:00:00.000Z"
                }
            },
            {
                "marketId": "1.113197507",
                "marketName": "Over/Under 4.5 Goals",
                "marketStartTime": "2014-03-13T11:00:00.000Z",
                "totalMatched": 3189.28,
                "runners": [
                    {
                        "selectionId": 1222347,
                        "runnerName": "Under 4.5 Goals",
                        "handicap": 0,
                        "sortPriority": 1,
                    "metadata": {
                        "runnerId": "63123524"
                    }
                },
                {
                    "selectionId": 1222346,
                    "runnerName": "Over 4.5 Goals",
                    "handicap": 0,
                    "sortPriority": 2,
                    "metadata": {
                        "runnerId": "63123525"
                    }
                }
            ],
            "eventType": {
                "id": "1",
                "name": "Soccer"
            },
            "competition": {
                "id": "2356065",
                "name": "Pirveli Liga"
            },
            "event": {
                "id": "27165685",
                "name": "FC Samtredia v FC Betlemi Keda",
                "countryCode": "GE",
                "timezone": "GMT",
                "openDate": "2014-03-13T11:00:00.000Z"
            }
        },
        {
            "marketId": "1.113197511",
            "marketName": "Over/Under 3.5 Goals",
            "marketStartTime": "2014-03-13T11:00:00.000Z",
            "totalMatched": 3934.41,
            "runners": [
                {
                    "selectionId": 1222344,
                    "runnerName": "Under 3.5 Goals",
                    "handicap": 0,
                    "sortPriority": 1,
                    "metadata": {
                        "runnerId": "63123547"
                    }
                },
                {
                    "selectionId": 1222345,
                    "runnerName": "Over 3.5 Goals",
                    "handicap": 0,
                    "sortPriority": 2,
                    "metadata": {
                        "runnerId": "63123548"
                    }
                }
            ],
            "eventType": {
                "id": "1",
                "name": "Soccer"
            },
            "competition": {
                "id": "2356065",
                "name": "Pirveli Liga"
            },
            "event": {
                "id": "27165685",
                "name": "FC Samtredia v FC Betlemi Keda",
                "countryCode": "GE",
                "timezone": "GMT",
                "openDate": "2014-03-13T11:00:00.000Z"
            }
        },
            {
                "marketId": "1.113197512",
                "marketName": "Asian Handicap",
                "marketStartTime": "2014-03-13T11:00:00.000Z",
                "totalMatched": 1599.38,
                "runners": [
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": -4,
                        "sortPriority": 1
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": 4,
                        "sortPriority": 2
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": -3.75,
                        "sortPriority": 3
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": 3.75,
                        "sortPriority": 4
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": -3.5,
                        "sortPriority": 5
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": 3.5,
                        "sortPriority": 6
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": -3.25,
                        "sortPriority": 7
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": 3.25,
                        "sortPriority": 8
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": -3,
                        "sortPriority": 9
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": 3,
                        "sortPriority": 10
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": -2.75,
                        "sortPriority": 11
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": 2.75,
                        "sortPriority": 12
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": -2.5,
                        "sortPriority": 13
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": 2.5,
                        "sortPriority": 14
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": -2.25,
                        "sortPriority": 15
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": 2.25,
                        "sortPriority": 16
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": -2,
                        "sortPriority": 17
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": 2,
                        "sortPriority": 18
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": -1.75,
                        "sortPriority": 19
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": 1.75,
                        "sortPriority": 20
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": -1.5,
                        "sortPriority": 21
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": 1.5,
                        "sortPriority": 22
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": -1.25,
                        "sortPriority": 23
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": 1.25,
                        "sortPriority": 24
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": -1,
                        "sortPriority": 25
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": 1,
                        "sortPriority": 26
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": -0.75,
                        "sortPriority": 27
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": 0.75,
                        "sortPriority": 28
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": -0.5,
                        "sortPriority": 29
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": 0.5,
                        "sortPriority": 30
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": -0.25,
                        "sortPriority": 31
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": 0.25,
                        "sortPriority": 32
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": 0,
                        "sortPriority": 33
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": 0,
                        "sortPriority": 34
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": 0.25,
                        "sortPriority": 35
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": -0.25,
                        "sortPriority": 36
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": 0.5,
                        "sortPriority": 37
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": -0.5,
                        "sortPriority": 38
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": 0.75,
                        "sortPriority": 39
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": -0.75,
                        "sortPriority": 40
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": 1,
                        "sortPriority": 41
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": -1,
                        "sortPriority": 42
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": 1.25,
                        "sortPriority": 43
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": -1.25,
                        "sortPriority": 44
                    },
                    {
                        "selectionId": 6830593,
                        "runnerName": "FC Samtredia",
                        "handicap": 1.5,
                        "sortPriority": 45
                    },
                    {
                        "selectionId": 6843866,
                        "runnerName": "FC Betlemi Keda",
                        "handicap": -1.5,
                        "sortPriority": 46
                    },
                    {
                        "selectionId": 6830593,
                    "runnerName": "FC Samtredia",
                    "handicap": 1.75,
                    "sortPriority": 47
                },
                {
                    "selectionId": 6843866,
                    "runnerName": "FC Betlemi Keda",
                    "handicap": -1.75,
                    "sortPriority": 48
                },
                {
                    "selectionId": 6830593,
                    "runnerName": "FC Samtredia",
                    "handicap": 2,
                    "sortPriority": 49
                },
                {
                    "selectionId": 6843866,
                    "runnerName": "FC Betlemi Keda",
                    "handicap": -2,
                    "sortPriority": 50
                },
                {
                    "selectionId": 6830593,
                    "runnerName": "FC Samtredia",
                    "handicap": 2.25,
                    "sortPriority": 51
                },
                {
                    "selectionId": 6843866,
                    "runnerName": "FC Betlemi Keda",
                    "handicap": -2.25,
                    "sortPriority": 52
                },
                {
                    "selectionId": 6830593,
                    "runnerName": "FC Samtredia",
                    "handicap": 2.5,
                    "sortPriority": 53
                },
                {
                    "selectionId": 6843866,
                    "runnerName": "FC Betlemi Keda",
                    "handicap": -2.5,
                    "sortPriority": 54
                },
                {
                    "selectionId": 6830593,
                    "runnerName": "FC Samtredia",
                    "handicap": 2.75,
                    "sortPriority": 55
                },
                {
                    "selectionId": 6843866,
                    "runnerName": "FC Betlemi Keda",
                    "handicap": -2.75,
                    "sortPriority": 56
                },
                {
                    "selectionId": 6830593,
                    "runnerName": "FC Samtredia",
                    "handicap": 3,
                    "sortPriority": 57
                },
                {
                    "selectionId": 6843866,
                    "runnerName": "FC Betlemi Keda",
                    "handicap": -3,
                    "sortPriority": 58
                },
                {
```

```
                "selectionId": 6830593,
                "runnerName": "FC Samtredia",
                "handicap": 3.25,
                "sortPriority": 59
            },
            {
                "selectionId": 6843866,
                "runnerName": "FC Betlemi Keda",
                "handicap": -3.25,
                "sortPriority": 60
            },
            {
                "selectionId": 6830593,
                "runnerName": "FC Samtredia",
                "handicap": 3.5,
                "sortPriority": 61
            },
            {
                "selectionId": 6843866,
                "runnerName": "FC Betlemi Keda",
                "handicap": -3.5,
                "sortPriority": 62
            },
            {
                "selectionId": 6830593,
                "runnerName": "FC Samtredia",
                "handicap": 3.75,
                "sortPriority": 63
            },
            {
                "selectionId": 6843866,
                "runnerName": "FC Betlemi Keda",
                "handicap": -3.75,
                "sortPriority": 64
            },
            {
                "selectionId": 6830593,
                "runnerName": "FC Samtredia",
                "handicap": 4,
                "sortPriority": 65,
                "metadata": {
                    "runnerId": "63123613"
                }
            },
            {
                "selectionId": 6843866,
                "runnerName": "FC Betlemi Keda",
                "handicap": -4,
                "sortPriority": 66,
                "metadata": {
                    "runnerId": "63123614"
                }
            }
        ],
        "eventType": {
            "id": "1",
            "name": "Soccer"
        },
        "competition": {
            "id": "2356065",
            "name": "Pirveli Liga"
        },
        "event": {
            "id": "27165685",
            "name": "FC Samtredia v FC Betlemi Keda",
            "countryCode": "GE",
            "timezone": "GMT",
            "openDate": "2014-03-13T11:00:00.000Z"
        }
    }
],
        "id": 1
    }
]
```

Horse Racing - Today's Win & Place Markets

The below request is an example of retrieving the available win/place horse racing markets for a specific day using listMarketCatalogue.
The marketStartTime (from & to) should be updated accordingly.

listMarketCatalogue Request

```
[
    {
        "jsonrpc": "2.0",
        "method": "SportsAPING/v1.0/listMarketCatalogue",
        "params": {
            "filter": {
                "eventTypeIds": [
                    "7"
                ],
                "marketTypeCodes": [
                    "WIN",
                    "PLACE"
                ],
                "marketStartTime": {
                    "from": "2013-10-16T00:00:00Z",
                    "to": "2013-10-16T23:59:00Z"
                }
            },
            "maxResults": "200",
            "marketProjection": [
                "MARKET_START_TIME",
                "RUNNER_METADATA",
                "RUNNER_DESCRIPTION",
                "EVENT_TYPE",
                "EVENT",
                "COMPETITION"
            ]
        },
        "id": 1
    }
]
```

Request a List of Football Competitions

To retrieve all football competitions, you can make a request to the listCompetitions operation with the following market filter:

listCompetitions Request

```
{
   "params": {
      "filter": {
         "eventTypeIds": [1]
      }
   },
   "jsonrpc": "2.0",
   "method": "SportsAPING/v1.0/listCompetitions",
   "id": 1
}
```

The "filter" selects all markets that have an eventTypeId of 1 (which is the event type for Football).

Then returns a list of Competitions and their Ids and how many markets are in each competition which are associated with those markets:

listCompetitions Response

```
{
   "jsonrpc": "2.0",
   "result": [
      {
         "marketCount": 16,
         "competition": {
            "id": "833222",
            "name": "Turkish Division 2"
         }
      },
      {
         "marketCount": 127,
         "competition": {
            "id": "5",
            "name": "A-League 2012/13"
         }
      },
      {
         "marketCount": 212,
         "competition": {
            "id": "7",
            "name": "Austrian Bundesliga"
         }
      },
      {
         "marketCount": 243,
         "competition": {
            "id": "11",
            "name": "Dutch Jupiler League"
         }
      },
      {
         "marketCount": 206,
         "competition": {
            "id": "26207",
            "name": "Greek Cup"
         }
      },
      {
         "marketCount": 117,
         "competition": {
            "id": "2129602",
            "name": "Professional Development League"
         }
      },
      {
         "marketCount": 68,
         "competition": {
            "id": "803237",
            "name": "Argentinian Primera B"
         }
      },
      {
         "marketCount": 1,
         "competition": {
            "id": "1842928",
            "name": "OTB Bank Liga"
         }
      }
   ],
   "id": 1
}
```

Python Example

```
import requests
import json

url="https://api.betfair.com/betting/json-rpc"
header = { 'X-Application' : "APP_KEY_HERE",  'X-Authentication : 'SESSION_TOKEN', 'content-type' : 'application
/json' }


jsonrpc_req='{"jsonrpc": "2.0", "method": "SportsAPING/v1.0/listCompetitions", "params": {"filter":{
"eventTypeIds" : [1]  }}, "id": 1}'


print json.dumps(json.loads(jsonrpc_req), indent=3)
print " "



response = requests.post(url, data=jsonrpc_req, headers=header)

print json.dumps(json.loads(response.text), indent=3)
```

Request Market Prices

Once you have identified the market (marketId) that your interested in using the listMarketCatalogue service, you can request prices for that market using the listMarketBook API call.

This is an example showing a request for the best prices (back and lay) and trading volume including virtual bets.

listMarketBook Request

```
[{
        "jsonrpc": "2.0",
        "method": "SportsAPING/v1.0/listMarketBook",
        "params": {
                "marketIds": ["1.127771425"],
                "priceProjection": {
                        "priceData": ["EX_BEST_OFFERS", "EX_TRADED"],
                        "virtualise": "true"
                }
        },
        "id": 1
}]
```

listMarketBook Response

```
[{
        "jsonrpc": "2.0",
        "result": [{
                "marketId": "1.127771425",
                "isMarketDataDelayed": false,
                "status": "OPEN",
                "betDelay": 0,
                "bspReconciled": false,
                "complete": true,
                "inplay": false,
                "numberOfWinners": 1,
                "numberOfRunners": 3,
                "numberOfActiveRunners": 3,
                "lastMatchTime": "2016-10-28T12:25:30.235Z",
                "totalMatched": 188959.22,
                "totalAvailable": 172932.96,
                "crossMatching": true,
                "runnersVoidable": false,
                "version": 1469071410,
                "runners": [{
                        "selectionId": 44790,
                        "handicap": 0.0,
                        "status": "ACTIVE",
                        "lastPriceTraded": 7.0,
                        "totalMatched": 12835.46,
                        "ex": {
                                "availableToBack": [{
                                        "price": 7.0,
                                        "size": 75.53
                                }, {
                                        "price": 6.8,
                                        "size": 538.6
                                }, {
                                        "price": 6.6,
                                        "size": 612.2
                                }],
                                "availableToLay": [{
                                        "price": 7.2,
                                        "size": 152.12
                                }, {
                                        "price": 7.4,
                                        "size": 1446.28
                                }, {
                                        "price": 7.6,
                                        "size": 1250.26
                                }],
                                "tradedVolume": [{
                                        "price": 6.4,
                                        "size": 3.0
                                }, {
                                        "price": 6.6,
                                        "size": 3.59
                                }, {
                                        "price": 6.8,
                                        "size": 1.42
                                }, {
                                        "price": 7.0,
                                        "size": 1736.55
                                }, {
                                        "price": 7.2,
                                        "size": 1601.31
                                }, {
                                        "price": 7.4,
                                        "size": 3580.1
                                }, {
                                        "price": 7.6,
                                        "size": 4236.59
                                }, {
                                        "price": 7.8,
                                        "size": 1367.18
                                }, {
                                        "price": 8.0,
                                        "size": 305.29
                                }, {
                                        "price": 8.2,
                                        "size": 0.39
                                }]
                        }
                }, {
                        "selectionId": 489720,
                        "handicap": 0.0,
                        "status": "ACTIVE",
                        "lastPriceTraded": 1.63,
                        "totalMatched": 163020.94,
                        "ex": {
                                "availableToBack": [{
                                        "price": 1.62,
                                        "size": 4921.06
                                }, {
                                        "price": 1.61,
                                        "size": 3230.34
                                }, {
                                        "price": 1.6,
                                        "size": 2237.71
                                }],
                                "availableToLay": [{
                                        "price": 1.63,
                                        "size": 1001.76
                                }, {
                                        "price": 1.64,
                                        "size": 6737.59
                                }, {
                                        "price": 1.65,
                                        "size": 1701.27
                                }],
                                "tradedVolume": [{
                                        "price": 1.53,
                                        "size": 31.38
                                }, {
                                        "price": 1.54,
                                        "size": 3.77
                                }, {
                                        "price": 1.57,
                                        "size": 3582.76
                                }, {
                                        "price": 1.58,
                                        "size": 12037.21
                                }, {
                                        "price": 1.59,
                                        "size": 16530.57
                                }, {
                                        "price": 1.6,
                                        "size": 54607.84
                                }, {
                                        "price": 1.61,
                                        "size": 36015.53
                                }, {
                                        "price": 1.62,
                                        "size": 21108.23
                                }, {
                                        "price": 1.63,
                                        "size": 17575.94
                                }, {
                                        "price": 1.64,
                                        "size": 1527.67
                                }]
                        }
                }, {
                        "selectionId": 58805,
                        "handicap": 0.0,
                        "status": "ACTIVE",
                        "lastPriceTraded": 4.2,
                        "totalMatched": 13102.81,
                        "ex": {
                                "availableToBack": [{
                                        "price": 4.1,
                                        "size": 3243.85
                                }, {
                                        "price": 4.0,
                                        "size": 1158.17
                                }, {
                                        "price": 3.95,
                                        "size": 254.04
                                }],
                                "availableToLay": [{
                                        "price": 4.2,
                                        "size": 1701.15
                                }, {
                                        "price": 4.3,
                                        "size": 3072.55
                                }, {
                                        "price": 4.4,
                                        "size": 2365.76
                                }],
                                "tradedVolume": [{
                                        "price": 4.0,
                                        "size": 457.79
                                }, {
                                        "price": 4.1,
                                        "size": 4434.67
                                }, {
                                        "price": 4.2,
                                        "size": 7845.01
                                }, {
                                        "price": 4.3,
                                        "size": 354.7
                                }, {
                                        "price": 4.4,
                                        "size": 6.6
                                }, {
                                        "price": 4.9,
                                        "size": 4.0
                                }]
                        }
                }]
        }],
        "id": 1
}]
```

Placing a Bet

To place a bet you require the marketId and selectionId parameters from the listMarketCatalogue API call. The below parameters will place a normal Exchange bet at odds of 3.0 for a stake of £2.0.

If the bet is placed successfully, a betId is returned in the placeOrders response

placeOrders Request

```
[
    {
        "jsonrpc": "2.0",
        "method": "SportsAPING/v1.0/placeOrders",
        "params": {
            "marketId": "1.109850906",
            "instructions": [
                {
                    "selectionId": "237486",
                    "handicap": "0",
                    "side": "LAY",
                    "orderType": "LIMIT",
                    "limitOrder": {
                        "size": "2",
                        "price": "3",
                        "persistenceType": "LAPSE"
                    }
                }
            ]
        },
        "id": 1
    }
]
```

placeOrder Response

```
[
    {
        "jsonrpc": "2.0",
        "result": {
            "marketId": "1.109850906",
            "instructionReports": [
                {
                    "instruction": {
                        "selectionId": 237486,
                        "handicap": 0,
                        "limitOrder": {
                            "size": 2,
                            "price": 3,
                            "persistenceType": "LAPSE"
                        },
                        "orderType": "LIMIT",
                        "side": "LAY"
                    },
                    "betId": "31242604945",
                    "placedDate": "2013-10-30T14:22:47.000Z",
                    "averagePriceMatched": 0,
                    "sizeMatched": 0,
                    "status": "SUCCESS"
                }
            ],
            "status": "SUCCESS"
        },
        "id": 1
    }
]
```

Placing a Betfair SP Bet - MARKET_ON_CLOSE

To place a bet on a selection at Betfair SP, you need to specify the parameters below in the placeOrders request. The below example would place a Betfair SP back bet on the required selection for a stake of £2.00.

Request

placeOrders Request

```
[
    {
        "jsonrpc": "2.0",
        "method": "SportsAPING/v1.0/placeOrders",
        "params": {
            "marketId": "1.111836557",
            "instructions": [
                {
                    "selectionId": "5404312",
                    "handicap": "0",
                    "side": "BACK",
                    "orderType": "MARKET_ON_CLOSE",
                    "marketOnCloseOrder": {
                        "liability": "2"
                    }
                }
            ]
        },
        "id": 1
    }
]
```

placeOrders Response

```
[
    {
        "jsonrpc": "2.0",
        "result": {
            "marketId": "1.111836557",
            "instructionReports": [
                {
                    "instruction": {
                        "selectionId": 5404312,
                        "handicap": 0,
                        "marketOnCloseOrder": {
                            "liability": 2
                        },
                        "orderType": "MARKET_ON_CLOSE",
                        "side": "BACK"
                    },
                    "betId": "31645233727",
                    "placedDate": "2013-11-12T12:07:29.000Z",
                    "status": "SUCCESS"
                }
            ],
            "status": "SUCCESS"
        },
        "id": 1
    }
]
```

Placing a Betfair SP Bet - LIMIT_ON_CLOSE

Below is an example of an Betfair SP Bet - using the orderType LIMIT_ON_CLOSE . Please refer to Additional Information#CurrencyParameters for Min BSP liability.

```
[{"jsonrpc": "2.0", "method": "SportsAPING/v1.0/placeOrders", "params": {"marketId":"1.148683414","
instructions":[{"selectionId":"15319829","handicap":"0","side":"LAY","orderType":"LIMIT_ON_CLOSE","
limitOnCloseOrder":{"price":"5","liability":"10"}}]}, "id": 1}
```

Retrieving Details of Bet/s Placed on a Market/s

You can use the listCurrentOrders request to retrieve of a bet/s placed on a specific market/s.

listCurrentOrders request

```
[{"jsonrpc": "2.0", "method": "SportsAPING/v1.0/listCurrentOrders", "params": {"marketIds":["1.117020524"],"
orderProjection":"ALL","dateRange":{}}, "id": 1}]
```

```
[{"jsonrpc":"2.0","result":{"currentOrders":[{"betId":"45496907354","marketId":"1.117020524","selectionId":
9170340,"handicap":0.0,"priceSize":{"price":10.0,"size":5.0},"bspLiability":0.0,"side":"BACK","status":"
EXECUTABLE","persistenceType":"LAPSE","orderType":"LIMIT","placedDate":"2015-01-22T13:01:53.000Z","
averagePriceMatched":0.0,"sizeMatched":0.0,"sizeRemaining":5.0,"sizeLapsed":0.0,"sizeCancelled":0.0,"
sizeVoided":0.0,"regulatorCode":"GIBRALTAR REGULATOR"}],"moreAvailable":false},"id":1}]
```

Retrieving the Result of a Settled Market

To retrieve the result of a settled market = make a request to listMarketBook **after** the market has been settled. The response will indicate whether the selection was settled as a 'WINNER' or 'LOSER' in the runners 'status' field. Settled market information is available for 90 days after settlement.

listMarketBook Request to a Settled market

```
[{"jsonrpc": "2.0", "method": "SportsAPING/v1.0/listMarketBook", "params": {"marketIds":["1.114363660"],"
priceProjection":{"priceData":["EX_BEST_OFFERS"]}}, "id": 1}]
```

listMarketBook Response

```
[
    {
        "jsonrpc": "2.0",
        "result": [
            {
                "marketId": "1.114363660",
                "isMarketDataDelayed": false,
                "status": "CLOSED",
                "betDelay": 0,
                "bspReconciled": false,
                "complete": true,
                "inplay": false,
                "numberOfWinners": 1,
                "numberOfRunners": 14,
                "numberOfActiveRunners": 0,
                "totalMatched": 0,
                "totalAvailable": 0,
                "crossMatching": false,
                "runnersVoidable": false,
                "version": 767398001,
                "runners": [
                    {
                        "selectionId": 8611526,
                        "handicap": 0,
                        "status": "REMOVED",
                        "adjustmentFactor": 9.1,
                        "removalDate": "2014-06-13T08:44:17.000Z",
                        "ex": {
                            "availableToBack": [],
                            "availableToLay": [],
                            "tradedVolume": []
                        }
                    },
                    {
                        "selectionId": 8611527,
                        "handicap": 0,
                        "status": "REMOVED",
                        "adjustmentFactor": 3.5,
                        "removalDate": "2014-06-13T08:44:29.000Z",
                        "ex": {
                            "availableToBack": [],
                            "availableToLay": [],
                            "tradedVolume": []
                        }
                    },
                    {
                        "selectionId": 7920154,
                        "handicap": 0,
                        "status": "WINNER",
                        "adjustmentFactor": 17.5,
                        "ex": {
                            "availableToBack": [],
                            "availableToLay": [],
                            "tradedVolume": []
                        }
                    },
                    {
                        "selectionId": 1231425,
                        "handicap": 0,
                        "status": "LOSER",
                        "adjustmentFactor": 4.3,
                        "ex": {
                            "availableToBack": [],
                            "availableToLay": [],
                            "tradedVolume": []
                        }
                    },
                    {
                        "selectionId": 7533866,
                        "handicap": 0,
                        "status": "LOSER",
                        "adjustmentFactor": 11.4,
                        "ex": {
                            "availableToBack": [],
                            "availableToLay": [],
                            "tradedVolume": []
                        }
                    },
                    {
                        "selectionId": 8220841,
                        "handicap": 0,
                        "status": "LOSER",
                        "adjustmentFactor": 5.4,
                        "ex": {
                            "availableToBack": [],
                            "availableToLay": [],
                            "tradedVolume": []
                        }
                    },
                    {
                        "selectionId": 7533883,
                        "handicap": 0,
                        "status": "LOSER",
                        "adjustmentFactor": 8.7,
                        "ex": {
                            "availableToBack": [],
                            "availableToLay": [],
                            "tradedVolume": []
                        }
                    },
                    {
                        "selectionId": 8476712,
                        "handicap": 0,
                        "status": "LOSER",
                        "adjustmentFactor": 8.7,
                        "ex": {
                            "availableToBack": [],
                            "availableToLay": [],
                            "tradedVolume": []
                        }
                    },
                    {
                        "selectionId": 7012263,
                        "handicap": 0,
                        "status": "LOSER",
                        "adjustmentFactor": 5.4,
                        "ex": {
                            "availableToBack": [],
                            "availableToLay": [],
                            "tradedVolume": []
                        }
                    },
                    {
                        "selectionId": 7374225,
                        "handicap": 0,
                        "status": "LOSER",
                        "adjustmentFactor": 8.7,
                        "ex": {
                            "availableToBack": [],
                            "availableToLay": [],
                            "tradedVolume": []
                        }
                    },
                    {
                        "selectionId": 8611525,
                        "handicap": 0,
                        "status": "LOSER",
                        "adjustmentFactor": 0.7,
                        "ex": {
                            "availableToBack": [],
                            "availableToLay": [],
                            "tradedVolume": []
                        }
                    },
                    {
                        "selectionId": 7659121,
                        "handicap": 0,
                        "status": "LOSER",
                        "adjustmentFactor": 26.8,
                        "ex": {
                            "availableToBack": [],
                            "availableToLay": [],
                            "tradedVolume": []
                        }
                    },
                    {
                        "selectionId": 6996332,
                        "handicap": 0,
                        "status": "LOSER",
                        "adjustmentFactor": 0.7,
                        "ex": {
                            "availableToBack": [],
                            "availableToLay": [],
                            "tradedVolume": []
                        }
                    },
                    {
                        "selectionId": 7137541,
                        "handicap": 0,
                        "status": "LOSER",
                        "adjustmentFactor": 1.7,
                        "ex": {
                            "availableToBack": [],
                            "availableToLay": [],
                            "tradedVolume": []
                        }
                    }
                ]
            }
        ],
        "id": 1
    }
]
```

Retrieving Details of Bets on a Settled Market - including P&L & Commission paid.

You can retrieve details of a settled bet/market using the listClearedOrders request. The below request uses a specific settledDatRange to limit the returned records and groupBy MARKET **rollup** to group results at markeId level.

listClearedOrders Request for bets on SETTLED markets

```
[{"jsonrpc": "2.0", "method": "SportsAPING/v1.0/listClearedOrders", "params": {"betStatus":"SETTLED","
settledDateRange":{"from":"2018-04-30T23:00:00Z","to":"2018-05-31T23:00:00Z"},"groupBy":"MARKET"}, "id": 1}]
```

Response

```
[{
        "jsonrpc": "2.0",
        "result": {
                "clearedOrders": [{
                        "eventTypeId": "1",
                        "eventId": "28746818",
                        "marketId": "1.144281274",
                        "placedDate": "2018-05-29T10:48:50.000Z",
                        "persistenceType": "LAPSE",
                        "orderType": "LIMIT",
                        "betOutcome": "LOST",
                        "settledDate": "2018-05-29T19:12:41.000Z",
                        "lastMatchedDate": "2018-05-29T15:25:26.000Z",
                        "betCount": 8,
                        "commission": 0.0,
                        "priceReduced": false,
                        "profit": -16.0
                }, {
                        "eventTypeId": "2",
                        "eventId": "28741289",
                        "marketId": "1.144157120",
                        "selectionId": 2830908,
                        "handicap": 0.0,
                        "betId": "126613673925",
                        "placedDate": "2018-05-25T08:26:20.000Z",
                        "persistenceType": "LAPSE",
                        "orderType": "LIMIT",
                        "side": "BACK",
                        "betOutcome": "WON",
                        "priceRequested": 1.28,
                        "settledDate": "2018-05-25T10:19:02.000Z",
                        "lastMatchedDate": "2018-05-25T08:26:40.000Z",
                        "betCount": 1,
                        "commission": 0.03,
                        "priceMatched": 1.28,
                        "priceReduced": false,
                        "sizeSettled": 2.0,
                        "profit": 0.56
                }, {
```

```
                "eventTypeId": "7",
                "eventId": "28739568",
                "marketId": "1.144114832",
                "selectionId": 11295111,
                "handicap": 0.0,
                "placedDate": "2018-05-23T21:27:52.000Z",
                "orderType": "LIMIT",
                "betOutcome": "WON",
                "settledDate": "2018-05-24T13:06:01.000Z",
                "lastMatchedDate": "2018-05-24T13:04:24.000Z",
                "betCount": 2,
                "commission": 0.11,
                "priceReduced": true,
                "profit": 2.14,
                "customerStrategyRef": "HedgerPro_216"
        }, {
                "eventTypeId": "7",
                "eventId": "28739518",
                "marketId": "1.144114517",
                "selectionId": 8912176,
                "handicap": 0.0,
                "betId": "126486489864",
                "placedDate": "2018-05-23T15:31:57.000Z",
                "persistenceType": "LAPSE",
                "orderType": "LIMIT",
                "side": "BACK",
                "betOutcome": "LOST",
                "priceRequested": 11.5,
                "settledDate": "2018-05-23T17:12:22.000Z",
                "lastMatchedDate": "2018-05-23T15:31:57.000Z",
                "betCount": 1,
                "commission": 0.0,
                "priceMatched": 11.5,
                "priceReduced": false,
                "sizeSettled": 2.65,
                "profit": -2.65
        }, {
                "eventTypeId": "1",
                "eventId": "28724382",
                "marketId": "1.143781550",
                "selectionId": 2081063,
                "handicap": 0.0,
                "placedDate": "2018-05-21T15:43:12.000Z",
                "persistenceType": "LAPSE",
                "orderType": "LIMIT",
                "side": "BACK",
                "betOutcome": "LOST",
                "priceRequested": 4.4,
                "settledDate": "2018-05-21T20:53:27.000Z",
                "lastMatchedDate": "2018-05-21T15:43:12.000Z",
                "betCount": 3,
                "commission": 0.0,
                "priceMatched": 4.4,
                "priceReduced": false,
                "sizeSettled": 6.0,
                "profit": -6.0
        }, {
                "eventTypeId": "1",
                "eventId": "28717428",
                "marketId": "1.143588270",
                "selectionId": 18565,
                "handicap": 0.0,
                "betId": "125772703658",
                "placedDate": "2018-05-16T14:48:53.000Z",
                "persistenceType": "LAPSE",
                "orderType": "LIMIT",
                "side": "BACK",
                "betOutcome": "WON",
                "priceRequested": 2.18,
                "settledDate": "2018-05-16T20:43:37.000Z",
                "lastMatchedDate": "2018-05-16T14:48:53.000Z",
                        "betCount": 1,
                        "commission": 0.12,
                        "priceMatched": 2.18,
                        "priceReduced": false,
                        "sizeSettled": 2.0,
                        "profit": 2.36
                }, {
                        "eventTypeId": "1",
                        "eventId": "28722989",
                        "marketId": "1.143732676",
                        "selectionId": 198689,
                        "handicap": 0.0,
                        "placedDate": "2018-05-14T13:22:56.000Z",
                        "persistenceType": "LAPSE",
                        "orderType": "LIMIT",
                        "side": "BACK",
                        "betOutcome": "WON",
                        "priceRequested": 1.42,
                        "settledDate": "2018-05-14T18:22:07.000Z",
                        "lastMatchedDate": "2018-05-14T13:22:56.000Z",
                        "betCount": 2,
                        "commission": 0.13,
                        "priceMatched": 1.42,
                        "priceReduced": false,
                        "sizeSettled": 6.0,
                        "profit": 2.52
                }],
                "moreAvailable": false
        },
        "id": 1
}]
```

Application Keys

What is an Application Key?

In order to use the Betting & Accounts API, you need to have an Application Key. The Application Key identifies your API client. Two App Keys are assigned to a single Betfair account, one **live** App Key and one **delayed** App Key for testing.

You must pass the Application Key with every HTTP request. You do this by setting the HTTP header with the value of the key assigned by Betfair.

```
X-Application: APP_KEY_ASSIGNED
```

Commercial Usage

Please note: App Key generation is for **personal betting** purposes only. All data/API usage in any commercial context must be approved by Betfair.

Unauthorised commercial usage will be identified & blocked.

How to Create An Application Key

You can create an Application Key for your Betfair account using the **Accounts API Demo Tool** and **createDeveloperAppKeys** operation

1. Click on the **Accounts API Demo Tool** link
2. Select the **createDeveloperAppKeys** operation from the list of Operations on the top left hand side of the demo tool.
3. Enter a **sessionToken** in the 'Session Token (ssoid)' text box. You can find instructions on how to find your sessionToken via your browser here.
4. Enter your **Application Name** (this must be unique) in the 'Request' column. The **Application Name** can be any name of your choice, but like your Betfair username, must be unique.
5. Press **Execute** at the bottom of the 'Request' column.

Two Application Keys will then be created and displayed in the **Developer Apps** column the demo tool

Please note:

- The X-Application header is not required when using the **createDeveloperAppKeys** or the **getDeveloperAppKeys** service.
- The **Application Name** must be unique.

We are aware that when using some browser versions to create App Keys the Demo Tool throws an **UNEXPECTED_ERROR** when requesting **createDeveloperAppKeys.** Using an alternative browser/s should resolve this problem.

You should also ensure that your Application Name is unique as attempts to create a duplicate Application Name will return an **APP_KEY_CREATION_FAILED** error response.

Live & Delayed Application Keys Usage

The **createDeveloperAppKeys** service will assign two Application Keys (App Keys) to your Betfair account.

One 'Live' Application Key and one 'Delayed' Application Key. A Delayed Application Key is displayed as 'Version 1.0-DELAY' via createDeveloperAppKeys/ getDeveloperAppKeys

Key points:

- Upon creation, the **Live Application Key will be inactive.**
- The **Delayed App Key** operates on the live (production) Betfair Exchange and **not** a **testbed/sandbox** environment.
- The **Delayed App Key** should be use for **development purposes** and ***any*** **functional testing**. The key provides **delayed Betfair price data.** The delay is variable between 1-180 second snapshots.
- The **Delayed App Key** must also be used in **simulation/practice applications** where the facility to bet into live Betfair markets is not available.
- The **Delayed App Key** does not return traded volume data '**totalMatched**' or **EX_ALL_OFFERS** via listMarketBook.

How do I activate my Live App Key?

To apply for a **Live Application key** please take note of the below:

Before applying, please:

- Complete any testing using your Delayed Application key.
- Ensure that your account has been fully verified in line with our KYC policy. **Please note:** We do not accept licence applications from **India, Bangledesh, Sri Lanka or the UAE**.
- Check that your account is funded to cover the £299 activation fee.

Please note: A one-off activation fee of **£299** applies; this is debited directly from your Betfair account once access is approved.

To apply for a **Live Application key** for personal betting use please click **here** and select **Exchange API > For My Personal Betting** and complete the application form at the bottom of the page.

Detailed **Historical Data** is additionally made available for testing and analysis purposes. Restrictions will be applied to 'read only' Live Application Keys.

Delay & Live Application Keys Overview

Please see below table for a summary of the data/services available to Delayed & Live Application Keys.

	Delayed Application Key	**Live Application Key**
Use For	**Development**	**Live betting applications**
Activation Fee	**None**	**£299**
Live Price Data	**Delayed***	**Yes**
Read Only Access Allowed	**Yes**	**No****
Bet Placement (Live Exchange)	**Yes**	**Yes**
Stream API	**Yes*****	**Yes**
Price Levels	**3**	**All**
Total Matched by Selection	**Not Available**	**Yes**
Total Matched by Market	**Yes**	**Yes**
BSP Far & Near Price	**Not Available**	**Yes**

`*Delay is variable between 1-180 seconds`
`**Restrictions will be applied to 'read only' Live App Keys`
`***Any new Delayed App Keys created since 8th April 2020`
`haved default access to Stream API.`
`Otherwise please contact Developer Support for access`		

Commercial Licensing

Please note: We do not accept licence applications from India, Bangledesh, Sri Lanka or the UAE

There are a number of different Commercial API licences available and these fit into the definitions below:

Software Vendor Licence

- **We wish to create a betting app to distribute to Betfair customers.**

Please see **Developer Support** for further information on how to apply for a Software Vendor Licence.

Odds Publisher Licence

- **We are a Betfair Affiliate & want to publish Betfair odds.**

If your not an Affiliate, you can apply via **https://affiliates.betfair.com/** **> Join Now**

Betting Operator Licence

- **We are a *licensed* Betting Operator wanting to use Exchange data.**

Please contact us via **Developer Support** for further information.

Login & Session Management

Login

The Betfair API offers three login flows for developers, depending on the use case of your application:.

Non-Interactive login

if you are building an application which will run autonomously, there is a separate login flow to follow to ensure your account remains secure.

Interactive login

if you are building an application which will be used interactively, then this is the flow for you. This flow has two variants:

Interactive login - Desktop Application

This login flow makes use of Betfair's login pages and allows your app to gracefully handle all errors and re-directions in the same way as the Betfair website

Interactive login - API method

This flow makes use of a JSON API Endpoint and is the simplest way to get started if you are looking to create your own login form.

If you're looking for the quickest way to get started, try the curl example in the Interactive login - API Method.

Login Request Limits

Successful login requests are restricted to **100 request per minute**.. In the event of a breach of the login limit the account will be prevented from creating new login session for a 20 minute period. The error **TEMPORARY_BAN_TOO_MANY_REQUESTS** will be returned in these circumstances. All existing sessions will continue to be valid.

Login FAQ's

When should I use the non-interactive login?

You should use the non-interactive login when the user will not be present to log into the application themselves. An example of this is an automated bot that might need to login without the user triggering a login. 3rd Party interfaces to Betfair, used by multiple users, and which act as a direct proxy of a user request should use the interactive login instead.

Why is the redirect URL required for the interactive login?

The redirect URL is required in order to post the session token to the application at the end of the login process. For further details of how to handle the session token please see Interactive Login from a Desktop

Why isn’t there a non-interactive endpoint that accepts only a username and a password?

Betfair take user security very seriously and have made many enhancements to the login process alongside additional changes which have been made at the request of some of our regulators. This means that you cannot rely upon a username and password being the only pieces of information that may be required at login. Some examples of workflows currently in use are 2 factor authorisation codes, additional National Identifiers for a region or requests for additional account information or account migration.

Why does my session time out, even though I’ve been retrieving prices?

For security reasons, we require that the application using the API explicitly calls the Keep Alive operation no more than once within every 8 hours in a response to user activity. In the case of non-interactive applications, these should call the keep-alive operation within every 8 hours whilst they are active.

Why is my interactive login/logout request failing with errorCode=FORBIDDEN?

Your Application Key App Key is not using the correct redirect URL. By default only https://www.betfair.com will be allowed as the redirect URL.

Keep Alive

You can use Keep Alive to extend the session timeout period. The minimum session time is currently **20 minutes (Italian Exchange)**. On the **international (.com) Exchange** the current session time is **8 hours**. Therefore, you should request Keep Alive within this time to prevent session expiry. If you don't call Keep Alive within the specified timeout period, the session will expire. **Please note:** Session times aren't determined or extended based on API activity.

Please note: You can configure the timeout via **My Account > Logout Preferences** if required

Headers

Name	Description	Sample
Accept (mandatory)	Header that signals that the response should be returned as JSON	application/json
X-Authentication (mandatory)	Header that represents the session token that needs to be keep alive	Session Token
X-Application (optional)	Header the Application Key used by the customer to identify the product.	App Key

ⓘ The presence of the "Accept: application/json" header will signal that the service should respond with JSON and not an HTML page

URL Definition (Global)

```
https://identitysso.betfair.com/api/keepAlive
```

Other Jurisdictions

Please use the below if your country of residence is in one of the list jurisdictions.

Jurisdiction	Endpoint
Australia	`https://identitysso.betfair.au/api/keepAlive`
Italy	`https://identitysso.betfair.it/api/keepAlive`
Spain	`https://identitysso.betfair.es/api/keepAlive`
Romania	`https://identitysso.betfair.ro/api/keepAlive`
Sweden	`https://identitysso.betfair.se/api/keepAlive`

Parameters

The Keep Alive operation requires no parameters.

Response structure

```
{
  "token":"<token_passed_as_header>",
  "product":"product_passed_as_header",
  "status":"<status>",
  "error":"<error>"
}
```

Status values

```
SUCCESS
FAIL
```

Error values

```
INPUT_VALIDATION_ERROR
INTERNAL_ERROR
NO_SESSION
```

Call sample

```
# full request
curl -k -i -H "Accept: application/json" -H "X-Application: AppKey" -H "X-Authentication: <token>" https://identitysso.betfair.com/api/keepAlive
```

You can use Keep Alive to extend the session timeout period. The minimum session time is currently 20 minutes (Italian Exchange). On the international (.com) Exchange the current session time is 8 hours. Therefore, you should request Keep Alive within this time to prevent session expiry. If you don't call Keep Alive within the specified timeout period, the session will expire. Session times aren't determined or extended based on API activity.

Keep Alive success

```
curl -k -i -H "Accept: application/json" -H "X-Application: AppKey" -H "X-Authentication: SESSIONTOKEN" https://identitysso.betfair.com/api/keepAlive

{
  "token":"SESSIONTOKEN",
  "product":"AppKey",
  "status":"SUCCESS",
  "error":""
}
```

Logout

You can use Logout to terminate your existing session.

URL Definition

`https://identitysso.betfair.com/api/logout`

The presence of the "Accept: application/json" header will signal that the service should respond with JSON and not an HTML page

Headers

Name	Description	Sample
Accept (mandatory)	Header that signals that the response should be returned as JSON	application/json
X-Authentication (mandatory)	Header that represents the session token created at login.	Session Token
X-Application (optional)	Header the Application Key used by the customer to identify the product.	App Key

Response structure

```
{
  "token":"<token_passed_as_header>",
  "product":"product_passed_as_header",
  "status":"<status>",
  "error":"<error>"
}
```

Status values

```
SUCCESS
FAIL
```

Error values

```
INPUT_VALIDATION_ERROR
INTERNAL_ERROR
NO_SESSION
```

Call sample

```
# full request
curl -k -i -H "Accept: application/json" -H "X-Application: AppKey" -H "X-Authentication: <token>" https://identitysso.betfair.com/api/logout
```

Non-Interactive (bot) login

The non-interactive login method for the Betfair API requires that you create and upload a self-signed certificate which will be used, alongside your username and password to authenticate your credentials and generate a session token.

For the purposes of this guide, we have used openssl to generate this client, details of which can be found at **http://www.openssl.org/**

2 Step Authentication With Non Interactive Login

Using **2 Step Authentication** to secure your account for website logins will have no impact on your use of the non-interactive login method and vice versa.

Getting Started

There are a couple of steps required before we can actually log in:

1. Create a self-signed certificate
2. Link the certificate to your Betfair account

Creating a Self Signed Certificate

API-NG requires that a 1024-bit or 2048-bit RSA certificate be used. There are various tutorials available on the Internet but be aware that the certificate needs to be for client authentication (most tutorials only cover server authentication).

Create a public/private RSA key pair using openssl

```
openssl genrsa -out client-2048.key 2048
```

Update or Create the openssl configuration file (openssl.cnf) for OpenSSL to override some of the default settings:

```
[ ssl_client ]
basicConstraints = CA:FALSE
nsCertType = client
keyUsage = digitalSignature, keyEncipherment
extendedKeyUsage = clientAuth
```

In Windows, the config file is located in the installation directory of OpenSSL

In Linux distributions, the config file is located at /usr/lib/ssl/openssl.cnf or /etc/ssl/openssl.cnf

Create a certificate signing request (CSR).

```
openssl req -new -config openssl.cnf -key client-2048.key -out client-2048.csr


Country Name (2 letter code) [AU]:GB
State or Province Name (full name) [Some-State]:London
Locality Name (eg, city) []:London
Organization Name (eg, company) [Internet Widgits Pty Ltd]:yourcompany.com
Organizational Unit Name (eg, section) []:Security Team
Common Name (e.g. server FQDN or YOUR name) []:Test API-NG Certificate
Email Address []:my.name@mydomain.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Self-sign the certificate request to create a certificate

```
openssl x509 -req -days 365 -in client-2048.csr -signkey client-2048.key -out client-2048.crt -extfile openssl.cnf -extensions ssl_client
```

In Windows, using any text editor, copy the contents of the .crt file and the .key file into a new file. Save this new file as client-2048.pem.

Linking the Certificate to Your Betfair Account

The previous steps should have created the following files:

File name	Description
client-2048.key	The private key. This file is needed in order to use the certificate and should be protected and shouldn't be shared with anyone.
client-2048.csr	A certificate signing request. This file is no longer needed and can be deleted.
client-2048.crt	The certificate. This file is not sensitive in security terms and can be shared with anyone.

Before you login using the certificate, it must be attached to your Betfair account, as follows:

1. Log in to your Betfair account through betfair.comPaste the following URL into the address bar of your browser
2. Navigate to https://myaccount.betfair.com/accountdetails/mysecurity?showAPI=1 - **Note:** Please use https://myaccount.betfair.it/accountdetails/mysecurity?showAPI=1 for the **Italian Exchange or the endpoint relevant to your own jusristiction. See the URL Definition section for more details**
3. Scroll to the section titled "**Automated Betting Program Access**" and click **'Edit'**
4. Click on "Browse" and then locate and select the file client-2048.crt (client-2048.pem if applicable) created above.
5. Click on the "**Upload Certificate**" button.

Scroll down to the "**Automated Betting Program Access**" section if required and the certificate details should be shown. You should now be able to log in to your Betfair account using the API-NG endpoint.

Note on File Formats

Some systems require that client certificates are in a different format to the ones we've created. The two most common formats are (a) PEM format key and certificate in a single file and (b) PKCS#12 format file. .NET applications require a PKCS#12 format file.

To create a PEM format file that contains both the private key and the certificate you can use the following command:

Linux

```
cat client-2048.crt client-2048.key > client-2048.pem
```

Create the PKCS#12 format using crt and key

```
openssl pkcs12 -export -in client-2048.crt -inkey client-2048.key -out client-2048.p12
```

Don't circulate the key, PEM file or PCKS#12 format files as these files are security sensitive

Details of a Login Request

A login request can now be made as follows:

1. Submit a HTTP “POST” request to: https://identitysso-cert.betfair.com/api/certlogin
2. As part of the SSL connection, the certificate created previously must be supplied.
3. Include a custom Header called “X-Application” with a value that identifies your application. The value is not validated and is only used to help with troubleshooting and diagnosing any problems.
4. Ensure the POST’s Content-Type is “application/x-www-form-urlencoded” rather than MIME attachment encoded.
5. As part of the POST body include two parameters “username” and “password” which should have the relevant username/password for your account.

Certificate Login Interface Details

URL Definition

Certificate Endpoint
`https://identitysso-cert.betfair.com/api/certlogin`

This endpoint is also available under the following jurisdictions

Please use the below if your country of residence is in one of the list jurisdictions.

Jurisdiction	Endpoint
Italy	https://identitysso-cert.betfair.it
Spain	https://identitysso-cert.betfair.es
Romania	https://identitysso-cert.betfair.ro
Sweden	https://identitysso-cert.betfair.se

Please note: Danish residents cannot use the Non-Interactive (bot) login method due to the NEMID requirement which is only supported by the Interactive Login - Desktop Application method

Request headers

- **X-Application** - You must set the X-Application header to your application key.

Request Parameters

- **username** (mandatory) - The username of the user logging in.
- **password** (mandatory) - The password of the user logging in.

Please note: The username and password values should be encoded when making the login request. All method names are case sensitive, this includes login, keepAlive and logout.

Response

The response returned is a json string. If the response is successful then the loginStatus key will contain SUCCESS, for example:

```
{
  sessionToken: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx;
  loginStatus: SUCCESS;
}
```

Should a failure or exception be returned, the response will be structured as below and loginStatus will contain a failure reason:

```
{
  loginStatus: INVALID_USERNAME_OR_PASSWORD;
}
```

The possible failure and exceptional return codes are:

error	Description
INVALID_USERNAME_OR_PASSWORD	the username or password are invalid
ACCOUNT_NOW_LOCKED	the account was just locked
ACCOUNT_ALREADY_LOCKED	the account is already locked
PENDING_AUTH	pending authentication
TELBET_TERMS_CONDITIONS_NA	Telbet terms and conditions rejected
DUPLICATE_CARDS	duplicate cards
SECURITY_QUESTION_WRONG_3X	the user has entered wrong the security answer 3 times
KYC_SUSPEND	KYC suspended
SUSPENDED	the account is suspended
CLOSED	the account is closed
SELF_EXCLUDED	the account has been self-excluded
INVALID_CONNECTIVITY_TO_REGULATOR_DK	the DK regulator cannot be accessed due to some internal problems in the system behind or in at regulator; timeout cases included.
NOT_AUTHORIZED_BY_REGULATOR_DK	the user identified by the given credentials is not authorized in the DK's jurisdictions due to the regulators' policies. Ex: the user for which this session should be created is not allowed to act(play, bet) in the DK's jurisdiction.
INVALID_CONNECTIVITY_TO_REGULATOR_IT	the IT regulator cannot be accessed due to some internal problems in the system behind or in at regulator; timeout cases included.
NOT_AUTHORIZED_BY_REGULATOR_IT	the user identified by the given credentials is not authorized in the IT's jurisdictions due to the regulators' policies. Ex: the user for which this session should be created is not allowed to act(play, bet) in the IT's jurisdiction.
SECURITY_RESTRICTED_LOCATION	the account is restricted due to security concerns
BETTING_RESTRICTED_LOCATION	the account is accessed from a location where betting is restricted
TRADING_MASTER	Trading Master Account
TRADING_MASTER_SUSPENDED	Suspended Trading Master Account
AGENT_CLIENT_MASTER	Agent Client Master
AGENT_CLIENT_MASTER_SUSPENDED	Suspended Agent Client Master
DANISH_AUTHORIZATION_REQUIRED	Danish authorization required
SPAIN_MIGRATION_REQUIRED	Spain migration required
DENMARK_MIGRATION_REQUIRED	Denmark migration required
SPANISH_TERMS_ACCEPTANCE_REQUIRED	The latest Spanish terms and conditions version must be accepted. You must login to the website to accept the new conditions.
ITALIAN_CONTRACT_ACCEPTANCE_REQUIRED	The latest Italian contract version must be accepted. You must login to the website to accept the new conditions.

CERT_AUTH_REQUIRED	Certificate required or certificate present but could not authenticate with it
CHANGE_PASSWORD_REQUIRED	Change password required
PERSONAL_MESSAGE_REQUIRED	Personal message required for the user
INTERNATIONAL_TERMS_ACCEPTANCE_REQUIRED	The latest international terms and conditions must be accepted prior to logging in.
EMAIL_LOGIN_NOT_ALLOWED	This account has not opted in to log in with the email
MULTIPLE_USERS_WITH_SAME_CREDENTIAL	There is more than one account with the same credential
ACCOUNT_PENDING_PASSWORD_CHANGE	The account must undergo password recovery to reactivate via https://identitysso.betfair.com/view/recoverpassword
TEMPORARY_BAN_TOO_MANY_REQUESTS	The limit for successful login requests per minute has been exceeded. New login attempts will be banned for 20 minutes
ITALIAN_PROFILING_ACCEPTANCE_REQUIRED	You must login to the website to accept the new conditions
AUTHORIZED_ONLY_FOR_DOMAIN_RO	You are attempting to login to the Betfair Romania domain with a non .ro account.
AUTHORIZED_ONLY_FOR_DOMAIN_SE	You are attempting to login to the Betfair Swedish domain with a non .se account.
SWEDEN_NATIONAL_IDENTIFIER_REQUIRED	You must provided your Swedish National identifier via Betfair.se before proceeding.
SWEDEN_BANK_ID_VERIFICATION_REQUIRED	You must provided your Swedish bank id via Betfair.se before proceeding.
ACTIONS_REQUIRED	You must login to https://www.betfair.com to provide missing information.

Sample curl command to quickly check the certificate based login

```
curl -q -k --cert client-2048.crt --key client-2048.key https://identitysso-cert.betfair.com/api/certlogin -d
"username=testuser&password=testpassword" -H "X-Application: curlCommandLineTest"

Response should be

{"sessionToken":"Zx8i4oigut5nc+l4L8qFb0DSxG+mwLn2t0AMGFxjrMJI=","loginStatus":"SUCCESS"}
```

Sample Code for Non-Interactive Login

Sample C# code using PKCS#12 key store

Please see code sample via https://github.com/betfair/API-NG-sample-code/tree/master/loginCode/Non-interactive-cSharp

Sample Java code using Apache http client library and PKCS#12 key store

Java API-NG login

```
package com.test.aping.client;



import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.conn.ClientConnectionManager;
import org.apache.http.conn.scheme.Scheme;
import org.apache.http.conn.ssl.SSLSocketFactory;
import org.apache.http.conn.ssl.StrictHostnameVerifier;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;

import javax.net.ssl.KeyManager;
import javax.net.ssl.KeyManagerFactory;
import javax.net.ssl.SSLContext;
import java.io.File;
import java.io.FileInputStream;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.SecureRandom;
import java.util.ArrayList;
import java.util.List;


public class HttpClientSSO {

    private static int port = 443;

    public static void main(String[] args) throws Exception {

        DefaultHttpClient httpClient = new DefaultHttpClient();

        try {
            SSLContext ctx = SSLContext.getInstance("TLS");
            KeyManager[] keyManagers = getKeyManagers("pkcs12", new FileInputStream(new File("C:
\\sslcerts\\client-2048.p12")), "password");
            ctx.init(keyManagers, null, new SecureRandom());
            SSLSocketFactory factory = new SSLSocketFactory(ctx, new StrictHostnameVerifier());

            ClientConnectionManager manager = httpClient.getConnectionManager();
            manager.getSchemeRegistry().register(new Scheme("https", port, factory));
            HttpPost httpPost = new HttpPost("https://identitysso-cert.betfair.com/api/certlogin");
            List<NameValuePair> nvps = new ArrayList<NameValuePair>();
            nvps.add(new BasicNameValuePair("username", "testuser"));
            nvps.add(new BasicNameValuePair("password", "testpassword"));

            httpPost.setEntity(new UrlEncodedFormEntity(nvps));



            httpPost.setHeader("X-Application","appkey");


            System.out.println("executing request" + httpPost.getRequestLine());

            HttpResponse response = httpClient.execute(httpPost);
            HttpEntity entity = response.getEntity();

            System.out.println("----------------------------------------");
            System.out.println(response.getStatusLine());
            if (entity != null) {
                String responseString = EntityUtils.toString(entity);
                                //extract the session token from responsestring
                                System.out.println("responseString" + responseString);
            }

        } finally {
            httpClient.getConnectionManager().shutdown();
        }
    }



    private static KeyManager[] getKeyManagers(String keyStoreType, InputStream keyStoreFile, String
keyStorePassword) throws Exception {
        KeyStore keyStore = KeyStore.getInstance(keyStoreType);
        keyStore.load(keyStoreFile, keyStorePassword.toCharArray());
        KeyManagerFactory kmf = KeyManagerFactory.getInstance(KeyManagerFactory.getDefaultAlgorithm());
        kmf.init(keyStore, keyStorePassword.toCharArray());
        return kmf.getKeyManagers();
    }
}
```

Sample Python code

```
#!/usr/bin/env python

import requests

#openssl x509 -x509toreq -in certificate.crt -out CSR.csr -signkey privateKey.key


payload = 'username=myusername&password=password'
headers = {'X-Application': 'SomeKey', 'Content-Type': 'application/x-www-form-urlencoded'}

resp = requests.post('https://identitysso-cert.betfair.com/api/certlogin', data=payload, cert=('client-2048.
crt', 'client-2048.key'), headers=headers)

if resp.status_code == 200:
  resp_json = resp.json()
  print resp_json['loginStatus']
  print resp_json['sessionToken']
else:
  print "Request failed."
```

Certificate Generation With XCA

If you aren't able (or willing) to setup openssl on your windows machine, there are various GUI wrappers around the toolset which you might be able to use instead. XCA is an open source wrapper around the OpenSSL toolset which allows you to create keys, csrs and certificates via a GUI and stores all of the generated items in a database file.

First you need to download XCA, which is an open source wrapper around the openssl toolset from:
http://sourceforge.net/projects/xca/

Step by Step Guide

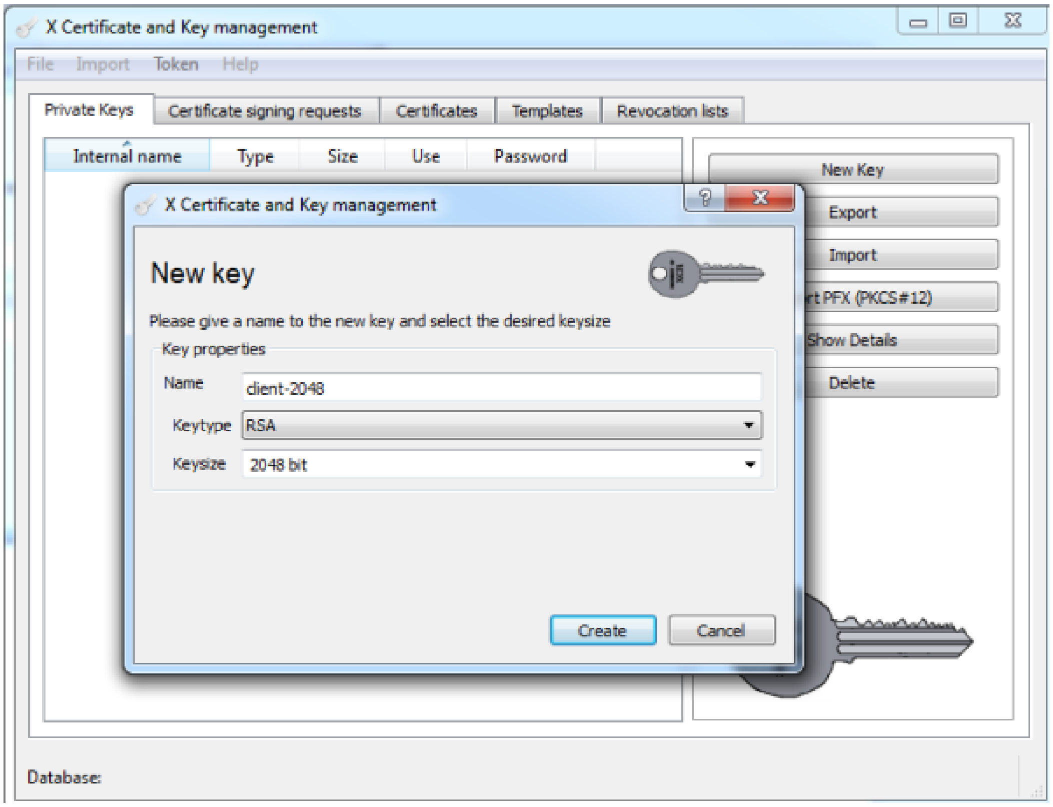
1. Install XCA and run it.
2. Create a new Database,
3. Name it something sensible
4. Save it somewhere appropriate.

This proprietary database is useful for the xca tool only and helps you store your keys, csrs, and certificates, the database file is not used in any part of the process with Betfair.

Equivalent Open SSL Command - Create a public/private RSA key pair using openssl

```
openssl genrsa -out client-2048.key 2048
```

- Create a public/private RSA key pair using xca:



X Certificate and Key management

New key

Please give a name to the new key and select the desired keysize

Key properties

Name: client-2048

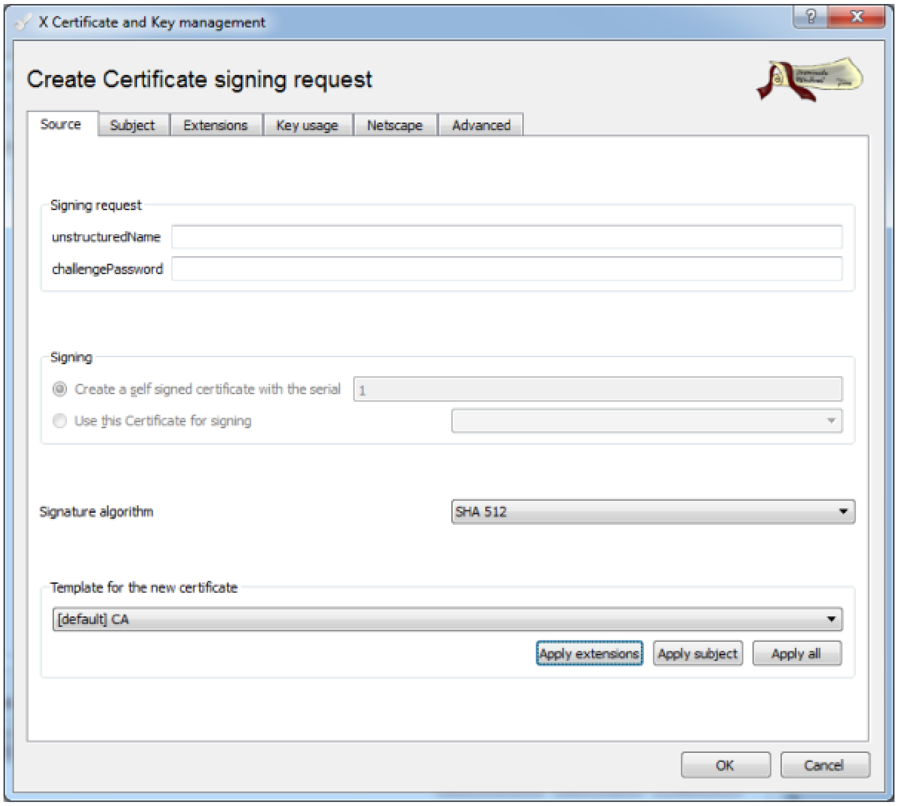
Keytype: RSA

Keysize: 2048 bit

Equivalent Open SSL Command - Create a certificate signing request (CSR).

```
openssl req -new -config openssl.cnf -key client-2048.key -out client-2048.csr
```

- Select the Certificate signing requests tab and click **New Request**
- Select the CA template and click the **Apply Extensions** button.



Create Certificate signing request

Signing request

unstructuredName

challengePassword

Signing

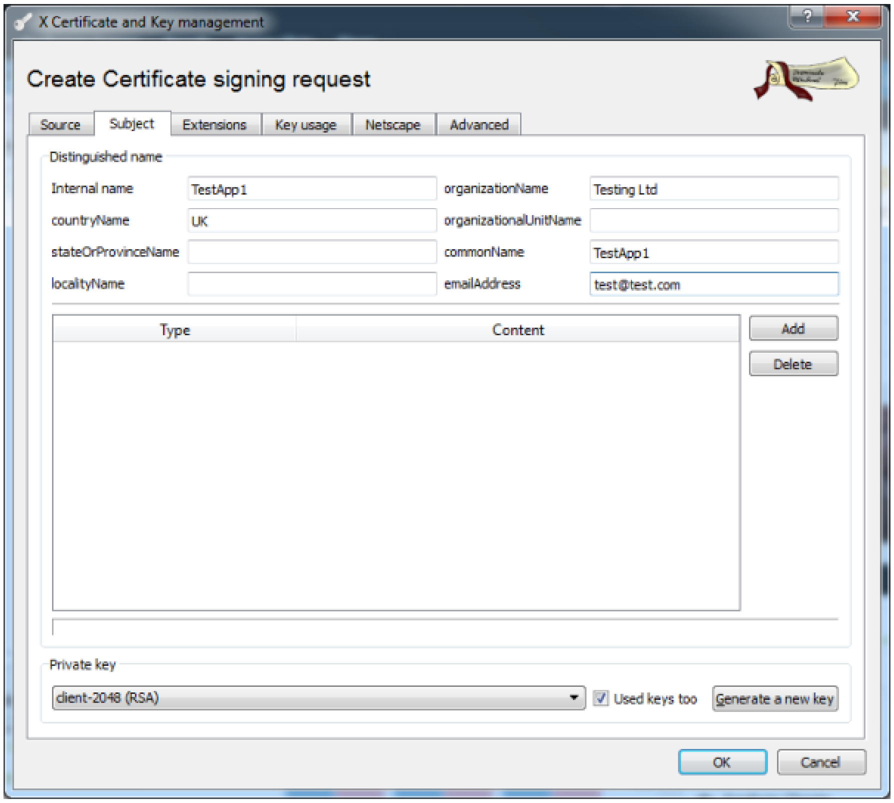
● Create a self signed certificate with the serial 1

○ Use this Certificate for signing

Signature algorithm: SHA 512

Template for the new certificate: [default] CA

- Click on the **Subject** tab and enter the name etc.
- The key that you generated in the first step must be selected (if the key doesn't appear, check the "Used keys tool" box)



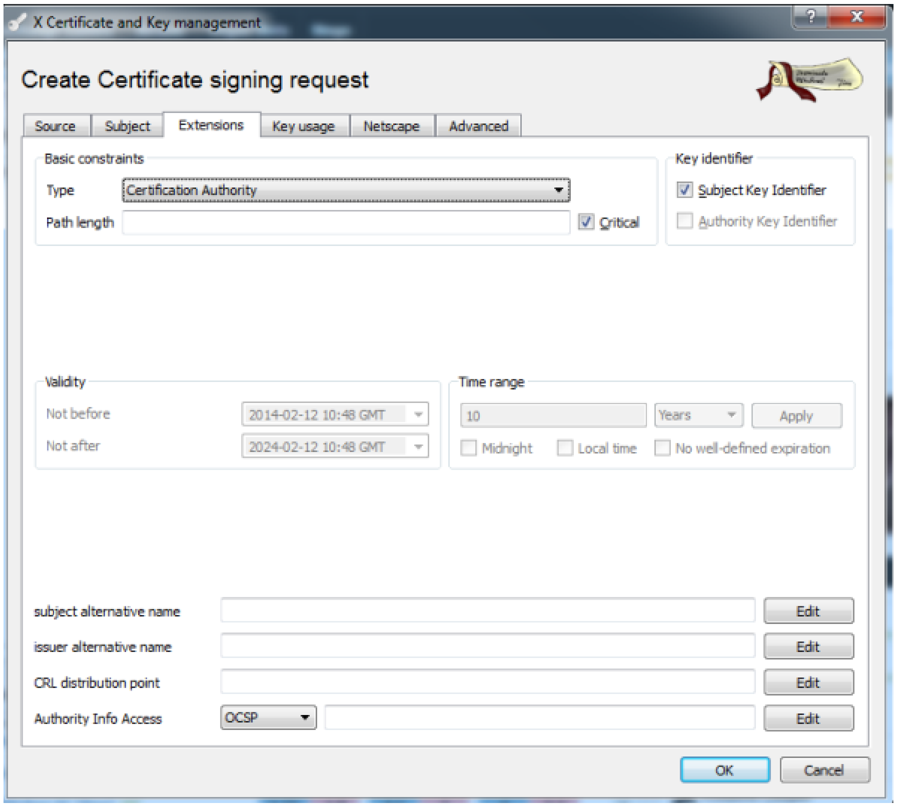
Create Certificate signing request

Distinguished name

Internal name	TestApp1	organizationName	Testing Ltd
countryName	UK	organizationalUnitName	
stateOrProvinceName		commonName	TestApp1
localityName		emailAddress	test@test.com

Private key: client-2048 (RSA) ☑ Used keys too

- Click on the **Extensions** tab
- Ensure that **Certification Authority** is selected as the type.
- Click OK.



X Certificate and Key management

Create Certificate signing request

Basic constraints

Type: Certification Authority

Path length:

☑ Critical

Key identifier

☑ Subject Key Identifier

□ Authority Key Identifier

Validity

Not before: 2014-02-12 10:48 GMT

Not after: 2024-02-12 10:48 GMT

Time range: 10 Years

□ Midnight □ Local time □ No well-defined expiration

subject alternative name:

issuer alternative name:

CRL distribution point:

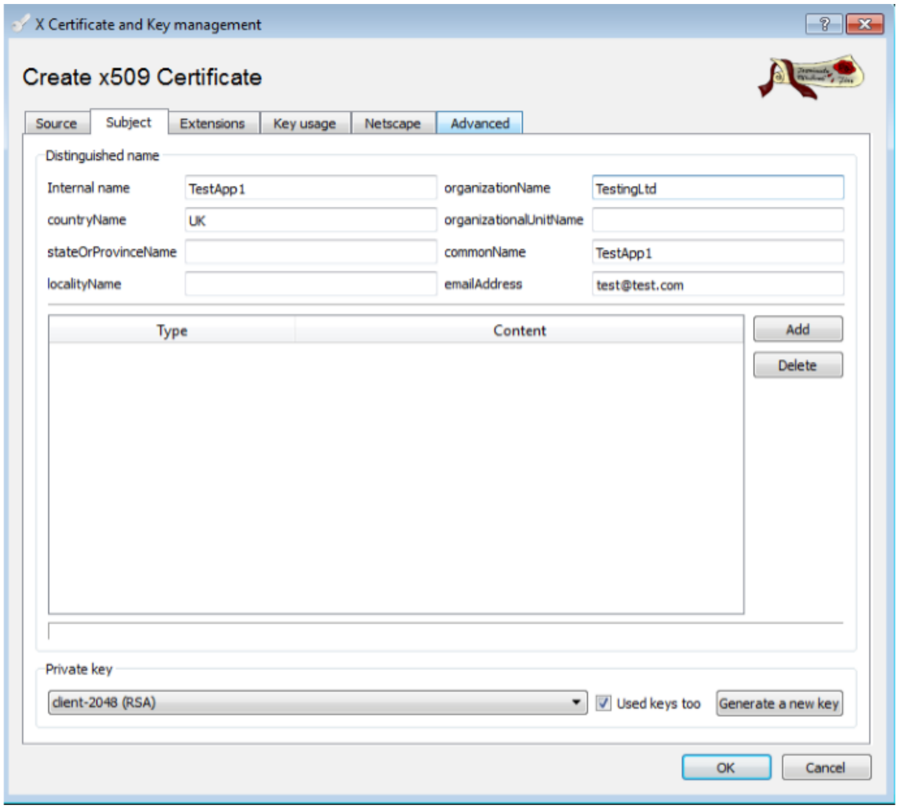
Authority Info Access: OCSP

Equivalent Open SSL Command - Self-sign the certificate request to create a certificate

```
openssl x509 -req -days 365 -in client-2048.csr -signkey client-2048.key -out client-2048.crt -extfile openssl.cnf -extensions ssl_client
```

We will now create and sign a certificate from the first two steps:

- Click the Certificates Tab and click New Certificate
- Click on the **Subject** tab and enter the name etc.
- The key that you generated in the first step must be selected (if the key doesn't appear, check the "Used keys tool" box)



Distinguished name

Internal name: TestApp1

countryName: UK

stateOrProvinceName:

localityName:

organizationName: TestingLtd

organizationalUnitName:

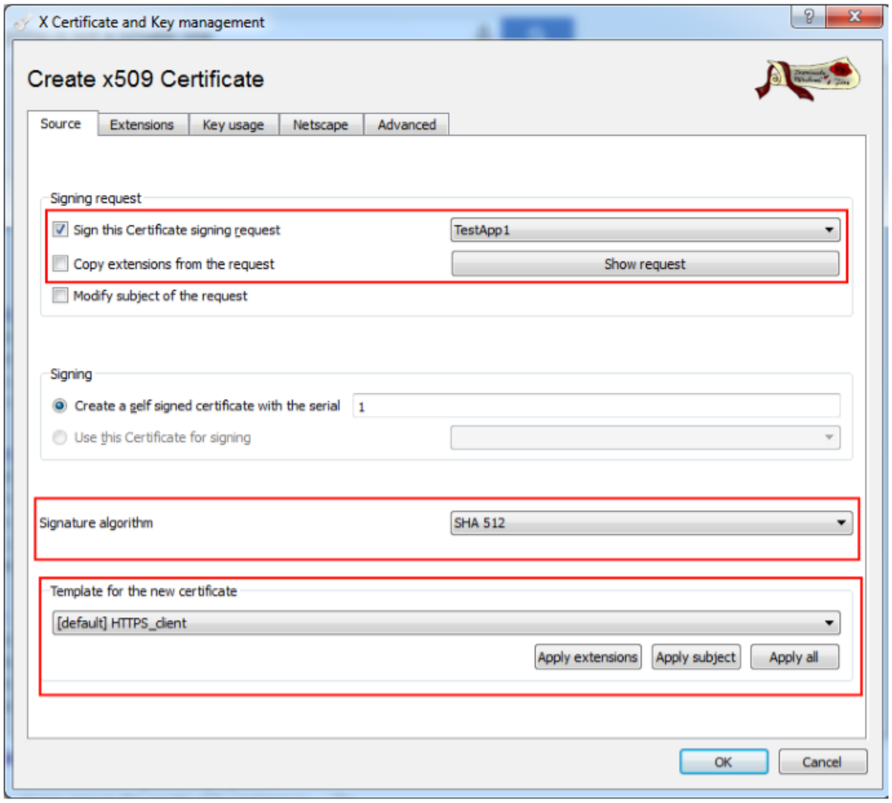
commonName: TestApp1

emailAddress: test@test.com

Private key: client-2048 (RSA)

☑ Used keys too

- Click on the **Source** tab and select the parameters shown below:



Signing request

☑ Sign this Certificate signing request: TestApp1

☐ Copy extensions from the request

☐ Modify subject of the request

Signing

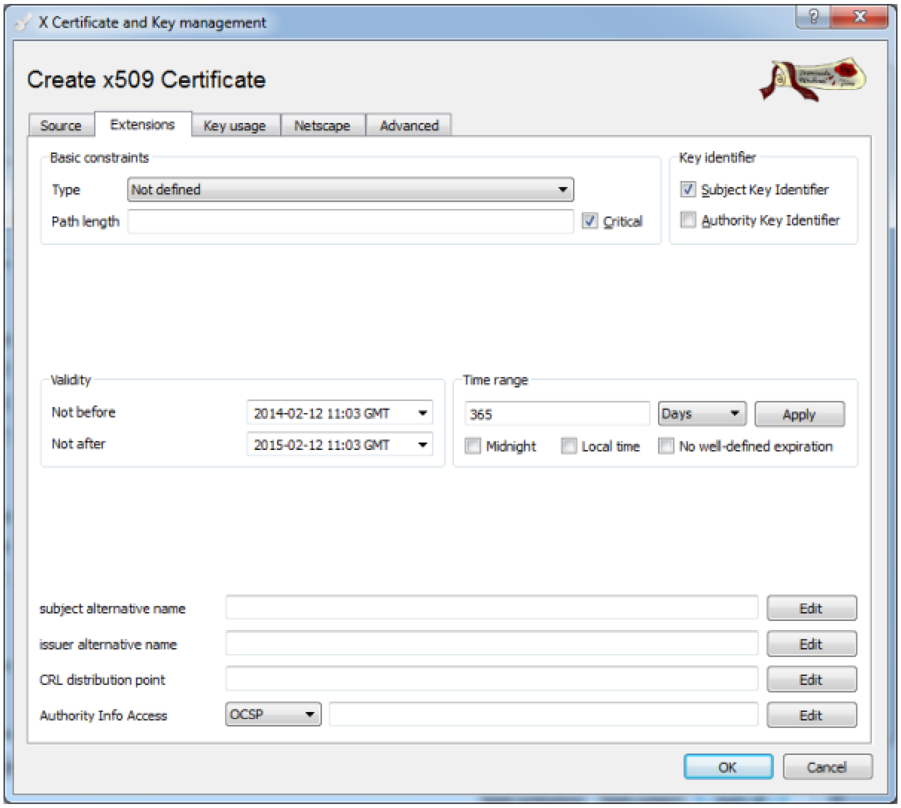
● Create a self signed certificate with the serial 1

○ Use this Certificate for signing

Signature algorithm: SHA 512

Template for the new certificate: [default] HTTPS_client

- You should make sure that your CSR is selected but that “Copy extensions from the request” is unticked and that you have selected the [default] HTTPS_client and pressed the “Apply extensions” button.



Create x509 Certificate

Basic constraints

Type: Not defined

Path length:

☑ Critical

Key identifier

☑ Subject Key Identifier

□ Authority Key Identifier

Validity

Not before: 2014-02-12 11:03 GMT

Not after: 2015-02-12 11:03 GMT

Time range

365 Days Apply

□ Midnight □ Local time □ No well-defined expiration

subject alternative name

issuer alternative name

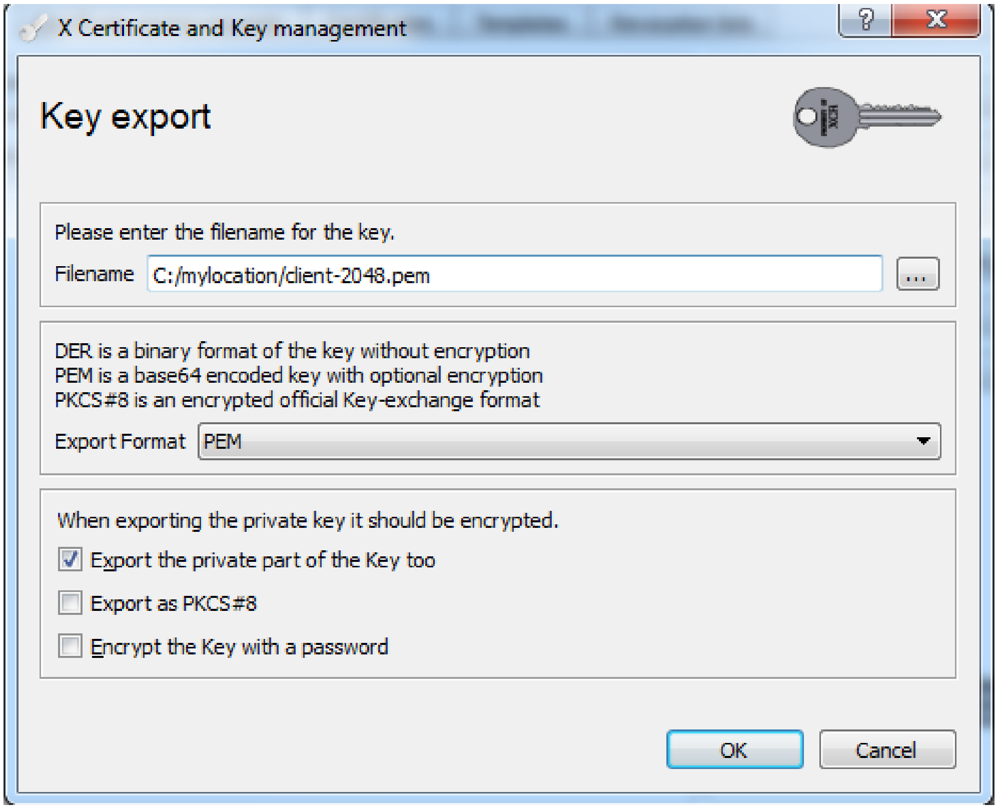
CRL distribution point

Authority Info Access: OCSP

You can then press OK to create the self-signed certificate.

Next we need to export the key and the certificate for use in our application.

- Export the private key and Certificate. These files should be used in the authentication process and should be referred to in your code where the private key (the .pem file) and certificate file (the .crt file) are mentioned. You can also export to other formats such as PKCS 12 depending upon the format expected by your language/library of choice.



X Certificate and Key management

Key export

Please enter the filename for the key.

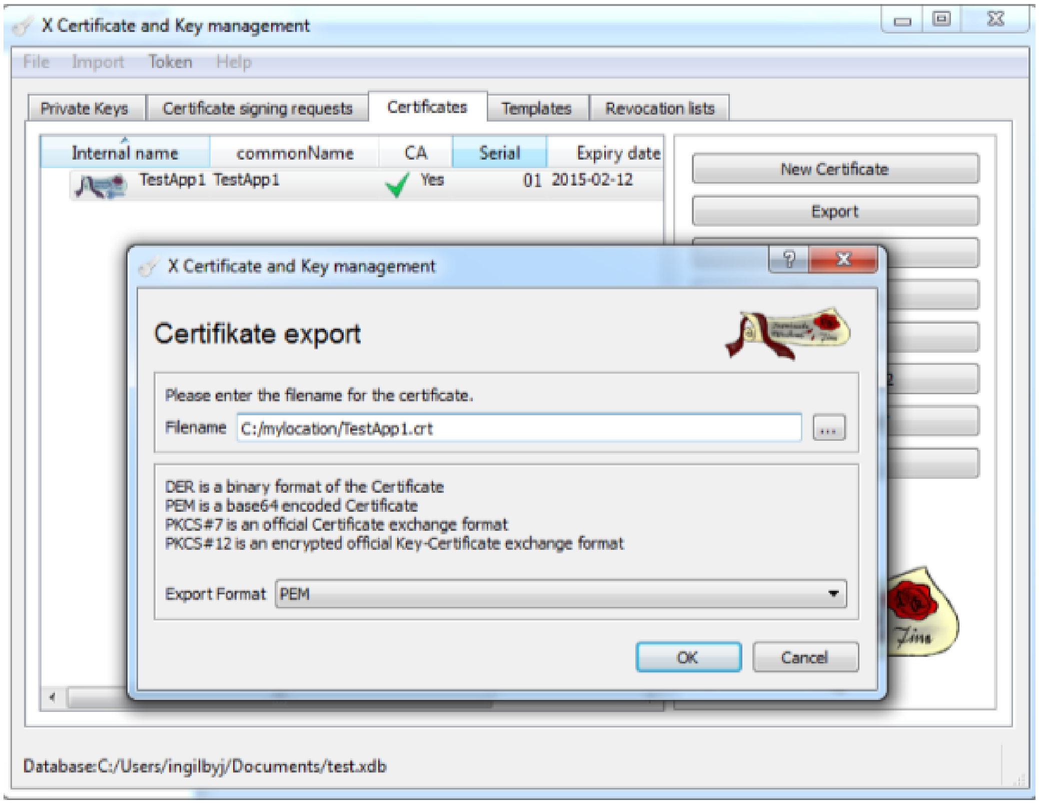
Filename: C:/mylocation/client-2048.pem

DER is a binary format of the key without encryption
PEM is a base64 encoded key with optional encryption
PKCS#8 is an encrypted official Key-exchange format

Export Format: PEM

When exporting the private key it should be encrypted.

- ☑ Export the private part of the Key too
- □ Export as PKCS#8
- □ Encrypt the Key with a password



X Certificate and Key management

Internal name	commonName	CA	Serial	Expiry date
TestApp1	TestApp1	✓ Yes	01	2015-02-12

X Certificate and Key management

Certifikate export

Please enter the filename for the certificate.

Filename: C:/mylocation/TestApp1.crt

DER is a binary format of the Certificate
PEM is a base64 encoded Certificate
PKCS#7 is an official Certificate exchange format
PKCS#12 is an encrypted official Key-Certificate exchange format

Export Format: PEM

Database:C:/Users/ingilbyj/Documents/test.xdb

You should upload the .crt file exported to the My Security page on Betfair.com to allow this certificate access to your account.

Interactive Login - Desktop Application

Overview

Interactive login is to be used when the user is present to login (for example, 3rd Party Desktop Applications) and will manage any additional information required at login depending upon a customer's account (such as 2 Factor Authentication codes or National Identifiers).

This is achieved by embedding the Betfair IdentitySSO login page in your application and then obtaining a successful session token upon login. The Keep Alive operation should be called within session expiry time if the user is still actively using your application.

Obtaining the sessionToken from the POST data

Once a login has been successfully made, the Javascript in the page will POST the session token (**ssoid**) to the URL provided as a redirect URL. For a desktop application, this is not required to be a real page as the desktop application can intercept the POST request as it happens via the embedded browser container. A Windows based application can embed a web browser into the application and use the BeforeNavigate2 event to catch the post data sent to the redirect URL and there are platform specific alternatives. The POST request body will contain two URL encoded parameters (which you will need to URL Decode):

- **ssoid** - This is your session token and should be attached to requests made to API-NG in the X-Authentication header.
- **errorCode** - This is returned in a URL by Betfair and provides the reason for the login failure.

This flow protects the implementing application from user login complexities, such as 2 factor auth, requiring national identifiers or jurisdictional migrations.

The Interactive Login is the same login flow used by the Betfair website and therefore, any message's will be returned directly by Betfair & handled in the same way.

errorCode	Description
INVALID_USERNAME_OR_PASSWORD	the username or password are invalid
ITALIAN_CONTRACT_ACCEPTANCE_REQUIRED	The latest italian contract version must be accepted
KYC_SUSPEND	KYC suspended
NOT_AUTHORIZED_BY_REGULATOR_DK	the user identified by the given credentials is not authorized in the DK's jurisdictions due to the regulators' policies. Ex: the user for which this session should be created is not allowed to act(play, bet) in the DK's jurisdiction.
NOT_AUTHORIZED_BY_REGULATOR_IT	the user identified by the given credentials is not authorized in the IT's jurisdictions due to the regulators' policies. Ex: the user for which this session should be created is not allowed to act(play, bet) in the IT's jurisdiction.
MULTIPLE_USERS_WITH_SAME_CREDENTIAL	There is more than one account with the same credential
PENDING_AUTH	pending authentication
PERSONAL_MESSAGE_REQUIRED	personal message required for the user
SECURITY_QUESTION_WRONG_3X	the user has entered wrong the security question 3 times
SECURITY_RESTRICTED_LOCATION	the account is restricted due to security concerns
SELF_EXCLUDED	the account has been self excluded
SPAIN_MIGRATION_REQUIRED	spain migration required
SPANISH_TERMS_ACCEPTANCE_REQUIRED	The latest spanish terms and conditions version must be accepted
STRONG_AUTH_CODE_RE	2 Step Authentication code is required. Please append this to your Betfair password.

QUIRED	
SUSPENDED	the account is suspended
TELBET_TERMS_CONDITIONS_NA	Telbet terms and conditions rejected
TRADING_MASTER	Trading Master Account
TRADING_MASTER_SUSPENDED	Suspended Trading Master Account
TEMPORARY_BAN_TOO_MANY_REQUESTS	The limit for successful login requests per minute has been exceeded. New login attempts will be banned for 20 minutes
INVALID_USERNAME_OR_PASSWORD	the username or password are invalid
ACCOUNT_NOW_LOCKED	the account was just locked
ACCOUNT_ALREADY_LOCKED	the account is already locked
PENDING_AUTH	pending authentication
TELBET_TERMS_CONDITIONS_NA	Telbet terms and conditions rejected
DUPLICATE_CARDS	duplicate cards
SECURITY_QUESTION_WRONG_3X	the user has entered wrong the security answer 3 times
KYC_SUSPEND	KYC suspended
SUSPENDED	the account is suspended
CLOSED	the account is closed
SELF_EXCLUDED	the account has been self-excluded
INVALID_CONNECTIVITY_TO_REGULATOR_DK	the DK regulator cannot be accessed due to some internal problems in the system behind or in at regulator; timeout cases included.
NOT_AUTHORIZED_BY_REGULATOR_DK	the user identified by the given credentials is not authorized in the DK's jurisdictions due to the regulators' policies. Ex: the user for which this session should be created is not allowed to act(play, bet) in the DK's jurisdiction.
INVALID_CONNECTIVITY_TO_REGULATOR_IT	the IT regulator cannot be accessed due to some internal problems in the system behind or in at regulator; timeout cases included.
NOT_AUTHORIZED_BY_REGULATOR_IT	the user identified by the given credentials is not authorized in the IT's jurisdictions due to the regulators' policies. Ex: the user for which this session should be created is not allowed to act(play, bet) in the IT's jurisdiction.
SECURITY_RESTRICTED_LOCATION	the account is restricted due to security concerns
BETTING_RESTRICTED_LOCATION	the account is accessed from a location where betting is restricted
TRADING_MASTER	Trading Master Account
TRADING_MASTER_SUSPENDED	Suspended Trading Master Account
AGENT_CLIENT_MASTER	Agent Client Master
AGENT_CLIENT_MASTER_SUSPENDED	Suspended Agent Client Master
DANISH_AUTHORIZATION_REQUIRED	Danish authorization required
SPAIN_MIGRATION_REQUIRED	Spain migration required
DENMARK_MIGRATION_REQUIRED	Denmark migration required
SPANISH_TERMS_ACCEPTANCE_REQUIRED	The latest Spanish terms and conditions version must be accepted. You must login to the website to accept the new conditions.

ITALIAN_CONTRACT_ACCEPTANCE_REQUIRED	The latest Italian contract version must be accepted. You must login to the website to accept the new conditions.
CERT_AUTH_REQUIRED	Certificate required or certificate present but could not authenticate with it
CHANGE_PASSWORD_REQUIRED	Change password required
PERSONAL_MESSAGE_REQUIRED	Personal message required for the user
INTERNATIONAL_TERMS_ACCEPTANCE_REQUIRED	The latest international terms and conditions must be accepted prior to logging in.
EMAIL_LOGIN_NOT_ALLOWED	This account has not opted in to log in with the email
MULTIPLE_USERS_WITH_SAME_CREDENTIAL	There is more than one account with the same credential
ACCOUNT_PENDING_PASSWORD_CHANGE	The account must undergo password recovery to reactivate via https://identitysso.betfair.com/view/recoverpassword
TEMPORARY_BAN_TOO_MANY_REQUESTS	The limit for successful login requests per minute has been exceeded. New login attempts will be banned for 20 minutes
ITALIAN_PROFILING_ACCEPTANCE_REQUIRED	You must login to the website to accept the new conditions
AUTHORIZED_ONLY_FOR_DOMAIN_RO	You are attempting to login to the Betfair Romania domain with a non .ro account.
AUTHORIZED_ONLY_FOR_DOMAIN_SE	You are attempting to login to the Betfair Swedish domain with a non .se account.
SWEDEN_NATIONAL_IDENTIFIER_REQUIRED	You must provided your Swedish National identifier via Betfair.se before proceeding.
SWEDEN_BANK_ID_VERIFICATION_REQUIRED	You must provided your Swedish bank id via Betfair.se before proceeding.
ACTIONS_REQUIRED	You must login to https://www.betfair.com to provide missing information.

URL Definition (Global)

```
https://identitysso.betfair.com/view/login?product=<theProductDescriptor>&url=<theRedirectUrl>
```

URL Definition - Other Jurisdictions

Please use the below if your country of residence is in one of the list jurisdictions.

Jurisdiction	Endpoint
Australia	https://identitysso.betfair.com.au/view/login?product=<theProductDescriptor>&url=<theRedirectUrl>
Italy	https://identitysso.betfair.it/view/login?product=<theProductDescriptor>&url=<theRedirectUrl>
Spain	`https://identitysso.betfair.es/view/login?product=<theProductDescriptor>&url=<theRedirectUrl>`
Romania	`https://identitysso.betfair.ro/view/login?product=<theProductDescriptor>&url=<theRedirectUrl>`
Sweden	`https://identitysso.betfair.se/view/login?product=<theProductDescriptor>&url=<theRedirectUrl>`

Parameters

Name	Description	Sample
product(mandatory)	The product for which the login page is used and on which the user will do the login; This should be your application key.	"IhDSui3ODdsdwo"

url (mandatory)	The url to which the the browser should be redirected in case of a successful login. By default only https://www.betfair.com will be allowed	https://www.betfair.com

Please note that all method names are case sensitive, this includes login, keepAlive and logout.

Interactive Login - API Endpoint

Overview and limitations

The API login endpoint is the simplest method of integration for most applications in terms of expected development time but comes at the cost of being less flexible to edge-cases than the embedded Betfair embedded login page. It will allow a user to provide a username and password or a username and (password + 2 factor auth code) if they have strong authentication enabled.

- Customers who writing bots are for their own use are **strongly recommended** to use the non-interactive endpoint with an SSL certificate.
- We **recommend** that **3rd party applications** which will be exposed to a wide range of users use the Interactive Login method of embedding the Betfair embedded login page as this will allow your application to handle additional workflows, such as terms and conditions updates as well as additional jurisdictional specific identifiers.

The Keep alive and logout methods remain the same with this method of login.

URL Definition (Global)

API Login Endpoint

```
https://identitysso.betfair.com/api/login
```

Other Jurisdictions

Please use the below if your country of residence is in one of the list jurisdictions.

Jurisdiction	Endpoint
Australia	`https://identitysso.betfair.com.au/api/login`
Italy	`https://identitysso.betfair.it/api/login`
Spain	`https://identitysso.betfair.es/api/login`
Romania	`https://identitysso.betfair.ro/api/login`
Sweden	`https://identitysso.betfair.se/api/login`

Parameters (POST)

Name	Description	Sample
username (mandatory)	The username to be used for the login	
password (mandatory)	The password to be used for the login. For strong auth customers, this should be their password with a 2 factor auth code appended to the password string.	

Headers

Name	Description	Sample
Accept (mandatory)	Signals that the response should be returned as JSON	application/json
X-Application (mandatory)	AppKey used by the customer to identify the product.	
Content-Type (required)	Required to support passwords containing special characters	application/x-www-form-urlencoded

The presence of the "Accept: application/json" will signal SSO that it should respond with JSON and not with a HTML page.

POST Example

Accept: application/json

X-Application: <AppKey>

Content-Type: application/x-www-form-urlencoded

URL endpoint: https://identitysso.betfair.com/api/login

Payload

username=username&password=password

Curl call sample

```
curl -k -i -H "Accept: application/json" -H "X-Application: <AppKey>" -X POST -d
'username=<username>&password=<password>' https://identitysso.betfair.com/api/login
```

Example of a successful login:

```
curl -k -i -H "Accept: application/json" -H "X-Application: <AppKey>" -X POST -d
'username=<username>&password=<password>' https://identitysso.betfair.com/api/login

{
  "token":"SESSION_TOKEN",
  "product":"APP_KEY",
  "status":"SUCCESS",
  "error":""
}
```

Response Structure

```
{
  "token":"<token_passed_as_header>",
  "product":"product_passed_as_header",
  "status":"<status>",
  "error":"<error>"
}
```

Status Values

```
SUCCESS
LIMITED_ACCESS
LOGIN_RESTRICTED
FAIL
```

Status Codes & Error values

The below describes the status codes that can be returned and the associated error values:

LIMITED_ACCESS - Access is limited (e.g. accounts can login but can't bet due to account suspension), product session will be provided.

```
{
  "token": product_token,
  "product": product,
  "status": LIMITED_ACCESS,
  "error": error
}

error = {PENDING_AUTH | SECURITY_QUESTION_WRONG_3X | KYC_SUSPEND | SUSPENDED}
```

LOGIN_RESTRICTED - login is restricted (in case of indirection point this is what will be returned), product session will not be provided:

```
{
  "token": "",
  "product": product,
  "status": LOGIN_RESTRICTED,
  "error": error
}

error = {STRONG_AUTH_CODE_REQUIRED | DENMARK_MIGRATION_REQUIRED | DANISH_AUTHORIZATION_REQUIRED |
SPAIN_MIGRATION_REQUIRED | SPANISH_TERMS_ACCEPTANCE_REQUIRED | ITALY_MIGRATION_REQUIRED |
ITALIAN_CONTRACT_ACCEPTANCE_REQUIRED | CHANGE_PASSWORD_REQUIRED | PERSONAL_MESSAGE_REQUIRED}
```

FAIL - All other cases are treated as errors, product session will not be provided:

```
{
  "token": "",
  "product": product,
  "status": FAIL,
  "error": error
}

error = {TRADING_MASTER | TRADING_MASTER_SUSPENDED | AGENT_CLIENT_MASTER | AGENT_CLIENT_MASTER_SUSPENDED |
DENMARK_MIGRATION_REQUIRED | INVALID_PIN | INVALID_USERNAME_OR_PASSWORD | PIN_DELETED_ON_FAILED_COUNT_EXCEEDED
| UNRECOGNIZED_DEVICE | DUPLICATE_CARDS | ACCOUNT_NOW_LOCKED | ACCOUNT_ALREADY_LOCKED |
SECURITY_RESTRICTED_LOCATION | BETTING_RESTRICTED_LOCATION | INVALID_CONNECTIVITY_TO_REGULATOR |
INVALID_CONNECTIVITY_TO_REGULATOR | INVALID_CONNECTIVITY_TO_REGULATOR_IT |
INVALID_CONNECTIVITY_TO_REGULATOR_DK| NOT_AUTHORIZED_BY_REGULATOR | NOT_AUTHORIZED_BY_REGULATOR |
NOT_AUTHORIZED_BY_REGULATOR_DK | NOT_AUTHORIZED_BY_REGULATOR_IT | TELBET_TERMS_CONDITIONS_NA | CLOSED |
SELF_EXCLUDED | NOT_AUTHORIZED_FOR_DOMAIN_ES | NOT_AUTHORIZED_FOR_DOMAIN_IT | NOT_AUTHORIZED_FOR_DOMAIN_COM |
AUTHORIZED_ONLY_FOR_DOMAIN_ES}
```

Please note that master account access is restricted for API/JSON requests.

```
{
  "token": "",
  "product": "APP_KEY",
  "status": FAIL,
  "error": error
}
```

```
error = {INPUT_VALIDATION_ERROR | FORBIDDEN | INVALID_USERNAME_OR_PASSWORD | NO_SESSION | INVALID_PIN |
INVALID_PIN_LOGIN_REQUEST | INVALID_PIN_LOGIN_REQUEST}
```

Possible failure and exceptional return codes

error	Description
ACCOUNT_ALREADY_LOCKED	the account is already locked
ACCOUNT_NOW_LOCKED	the account was just locked
ACCOUNT_PENDING_PASSWORD_CHANGE	The account must undergo password recovery to reactivate via https://identitysso.betfair.com/view/recoverpassword
ACTIONS_REQUIRED	You must login to https://www.betfair.com to provide missing information.
AGENT_CLIENT_MASTER	Agent Client Master
AGENT_CLIENT_MASTER_SUSPENDED	Suspended Agent Client Master
AUTHORIZED_ONLY_FOR_DOMAIN_RO	You are attempting to login to the Betfair Romania domain with a non .ro account.
AUTHORIZED_ONLY_FOR_DOMAIN_SE	You are attempting to login to the Betfair Swedish domain with a non .se account.
BETTING_RESTRICTED_LOCATION	the account is accessed from a location where betting is restricted
CERT_AUTH_REQUIRED	Certificate required or certificate present but could not authenticate with it
CHANGE_PASSWORD_REQUIRED	Change password required
CLOSED	the account is closed
DANISH_AUTHORIZATION_REQUIRED	Danish authorization required
DENMARK_MIGRATION_REQUIRED	Denmark migration required
DUPLICATE_CARDS	duplicate cards
EMAIL_LOGIN_NOT_ALLOWED	This account has not opted in to log in with the email
INTERNATIONAL_TERMS_ACCEPTANCE_REQUIRED	The latest international terms and conditions must be accepted prior to logging in.
INVALID_CONNECTIVITY_TO_REGULATOR_DK	the DK regulator cannot be accessed due to some internal problems in the system behind or in at regulator; timeout cases included.
INVALID_CONNECTIVITY_TO_REGULATOR_IT	the IT regulator cannot be accessed due to some internal problems in the system behind or in at regulator; timeout cases included.
INVALID_USERNAME_OR_PASSWORD	the username or password are invalid
ITALIAN_CONTRACT_ACCEPTANCE_REQUIRED	The latest Italian contract version must be accepted. You must login to the website to accept the new conditions.
ITALIAN_PROFILING_ACCEPTANCE_REQUIRED	You must login to the website to accept the new conditions
KYC_SUSPEND	KYC suspended
MULTIPLE_USERS_WITH_SAME_CREDENTIAL	There is more than one account with the same credential
NOT_AUTHORIZED_BY_REGULATOR_DK	the user identified by the given credentials is not authorized in the DK's jurisdictions due to the regulators' policies. Ex: the user for which this session should be created is not allowed to act(play, bet) in the DK's jurisdiction.
NOT_AUTHORIZED_BY_RE	the user identified by the given credentials is not authorized in the IT's jurisdictions due to the regulators' policies. Ex:

GULATOR_IT	the user for which this session should be created is not allowed to act(play, bet) in the IT's jurisdiction.
PENDING_AUTH	pending authentication
PERSONAL_MESSAGE_REQUIRED	Personal message required for the user
SECURITY_QUESTION_WRONG_3X	the user has entered wrong the security answer 3 times
SECURITY_RESTRICTED_LOCATION	the account is restricted due to security concerns
SELF_EXCLUDED	the account has been self-excluded
SPAIN_MIGRATION_REQUIRED	Spain migration required
SPANISH_TERMS_ACCEPTANCE_REQUIRED	The latest Spanish terms and conditions version must be accepted. You must login to the website to accept the new conditions.
SUSPENDED	the account is suspended
SWEDEN_BANK_ID_VERIFICATION_REQUIRED	You must provided your Swedish bank id via Betfair.se before proceeding.
SWEDEN_NATIONAL_IDENTIFIER_REQUIRED	You must provided your Swedish National identifier via Betfair.se before proceeding.
TELBET_TERMS_CONDITIONS_NA	Telbet terms and conditions rejected
TEMPORARY_BAN_TOO_MANY_REQUESTS	The limit for successful login requests per minute has been exceeded. New login attempts will be banned for 20 minutes
TRADING_MASTER	Trading Master Account
TRADING_MASTER_SUSPENDED	Suspended Trading Master Account

Best Practice

Development & Testing

You should use the **Delayed Application Key** for any initial development and functional testing. Historical data is made available via **https://historicdata.betfair.com/#/home** for strategy modelling & analysis.

Only apply for Live Application Key access once you are ready to start transacting on the Exchange using your Live Application Key.

Please see the **Personal Betting Access Overview** for more details regarding the difference between Delayed an Live Application Keys.

Session Management

Use Login to create a new session and Keep Alive to extend the session beyond the stated session expiry time. A single session can be used across multiple API calls/threads simultaneously.

You should ensure that you handle the INVALID_SESSION_TOKEN error within your code by creating a new session token via the API login method.

General Tips

- Make the minimal number of transactions/changes possible when transacting.
- Observe the Market Data Limits when making requests to listMarketCatalogue, listRunnerBook, listMarketBook and listMarketProfitandLoss
- Always prefer leaving an order in place rather than cancelling/re-placing it – stay at the front of the queue to be matched!
- Use the Stream API instead of polling wherever possible, particularly if you are running a high frequency trading application.
- Log as much as possible to aid queries/problem investigation (especially connectionid when using the Stream API)
- Make use of betting enhancements

API Status

Use the API status page http://status.developer.betfair.com/ to check the health of the API.

The API Status:

- Measures response latency and error rate against a number of operations every second
- Automatically toggles the status page if certain thresholds are breached

Please check the API status before contacting Developer Support regarding API problems.

Expect: 100 - Continue Header

Sending this header will result in the error: **"The remote server returned an error: (417) Expectation Failed."**

You should be aware that if using the .Net Framework you will need to set the relevant property in the ServicePointManager which then prevents the "Expect" header from being added:

System.Net.ServicePointManager.Expect100Continue = false;

Enabling HTTP Compression

HTTP compression is a capability built into both web servers and web clients to reduce the number of bytes transmitted in an HTTP response. This makes better use of available bandwidth and increases performance while reducing download time. When enabled, HTTP protocol data is compressed before it is sent from the server. Clients capable of receiving compressed HTTP data announce that they support compression in the HTTP header. Almost all modern web browsers support HTTP Compression by default.

The Betfair API uses HTTP to handle communication between API clients and servers. Therefore, the JSON messages can be compressed using the same HTTP compression used by web browsers. Custom API applications may need some modification before they can take advantage of this feature. Specifically, they need to send an additional HTTP header to indicate they support receipt of compressed responses from the API. In addition, some environments require you to explicitly decompress the response.

We would therefore recommend that all Betfair API request are sent with the ‘**Accept-Encoding: gzip, deflate’** request header.

HTTP Persistent Connection

We recommend that **Connection: keep-alive** header is set for all requests to guarantee a persistent connection and therefore reducing latency. **Please note**: Idle keep-alive connection to the API endpoints are closed every 3 minutes.

Other Performance Tips

Additional advice regarding optimizing HTTPClient performance can be found via **Connection management**

API Demo Tools

What are Demo Tools?

Demo tools are available for the Betfair Exchange API for both the **Betting** and **Accounts** API. The Demo Tools can be used by developers to allow quick experimentation and interaction with the **production API system.**

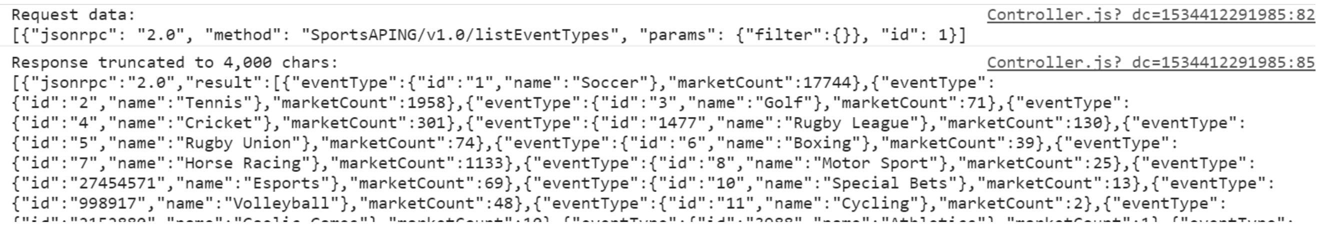
Betting API Demo tool - https://docs.developer.betfair.com/visualisers/api-ng-sports-operations/

Accounts API Demo tool - https://docs.developer.betfair.com/visualisers/api-ng-account-operations/

Why Use Demo Tools?

- All existing API operations are available
- **Easily experiment** with the parameters for your API queries.
- **Build sample requests** and interact with the API directly requests and responses are output to the JavaScript debug console.

  For example, using Google Chrome, you can open the Javascript console using the **shortcut Ctrl+Shift+J** to display output as below. The same shortcut can also be used with Mozilla Firefox.



```
Request data:                                                                          Controller.js? dc=1534412291985:82
[{"jsonrpc": "2.0", "method": "SportsAPING/v1.0/listEventTypes", "params": {"filter":{}}, "id": 1}]

Response truncated to 4,000 chars:                                                     Controller.js? dc=1534412291985:85
[{"jsonrpc":"2.0","result":[{"eventType":{"id":"1","name":"Soccer"},"marketCount":17744},{"eventType":
{"id":"2","name":"Tennis"},"marketCount":1958},{"eventType":{"id":"3","name":"Golf"},"marketCount":71},{"eventType":
{"id":"4","name":"Cricket"},"marketCount":301},{"eventType":{"id":"1477","name":"Rugby League"},"marketCount":130},{"eventType":
{"id":"5","name":"Rugby Union"},"marketCount":74},{"eventType":{"id":"6","name":"Boxing"},"marketCount":39},{"eventType":
{"id":"7","name":"Horse Racing"},"marketCount":1133},{"eventType":{"id":"8","name":"Motor Sport"},"marketCount":25},{"eventType":
{"id":"27454571","name":"Esports"},"marketCount":69},{"eventType":{"id":"10","name":"Special Bets"},"marketCount":13},{"eventType":
{"id":"998917","name":"Volleyball"},"marketCount":48},{"eventType":{"id":"11","name":"Cycling"},"marketCount":2},{"eventType":
{"id":"2152880","name":"Gaelic Games"},"marketCount":10},{"eventType":{"id":"3988","name":"Athletics"},"marketCount":1},{"eventType":
```

Please note that the Demo Tools are provided as-is, and are intended solely as a testing resource

Obtaining a Session Token for the Demo Tools

There are several ways that you can obtain a session token for the visualiser:

Using the pre-populated session token from the website

The Demo Tools will populate the session token field with the value of a session token found in your browsers cookie store for the Betfair.com website. Login to www.betfair.com (via a seperate browser tab) and refresh the demo tools page to automatically input the Session Token field

Manually adding the ssoid (session token) from the website cookie

Using Google Chrome you can inspect and copy the session directly from the browser by doing the following:

1, Press Ctrl+Shift+J
2. Select Application > Cookies > **www.betfair.com or www.betfair.com.au**
3. Copy the Value for the cookie with name **ssoid** and paste this into the Demo Tool

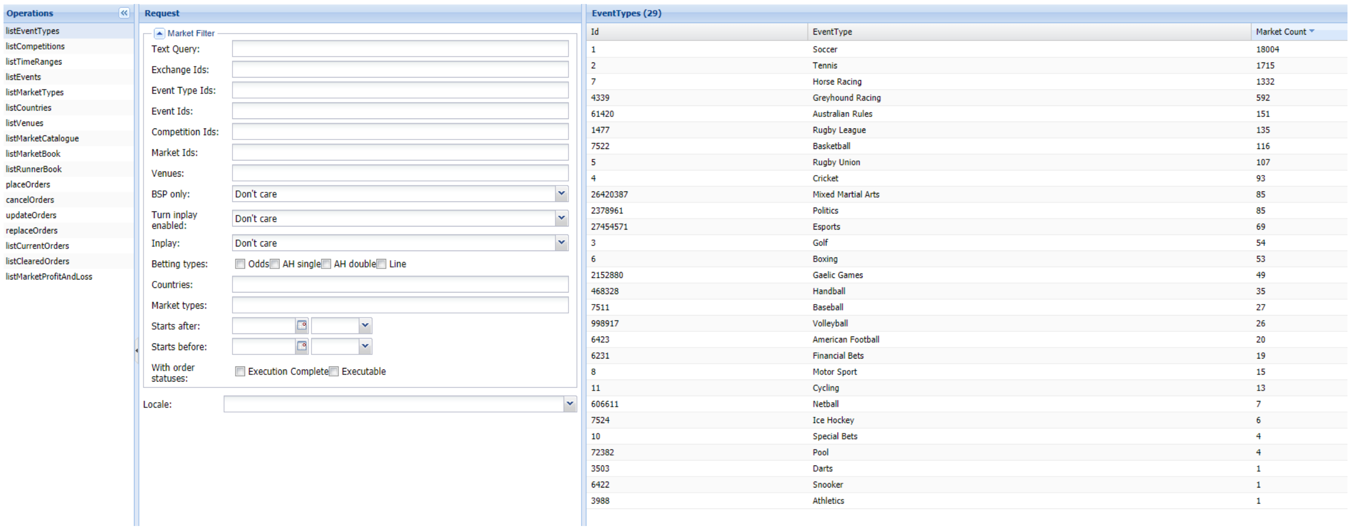
Login using the API Endpoint login request

Make a POST request using a tool such as Postman following the instructions via **Interactive Login - API Endpoint**

Examples of Demo Tools Usage

Please see some example of how to use the Demo tools in the steps below:

1. Request All eventTypes using listEventTypes

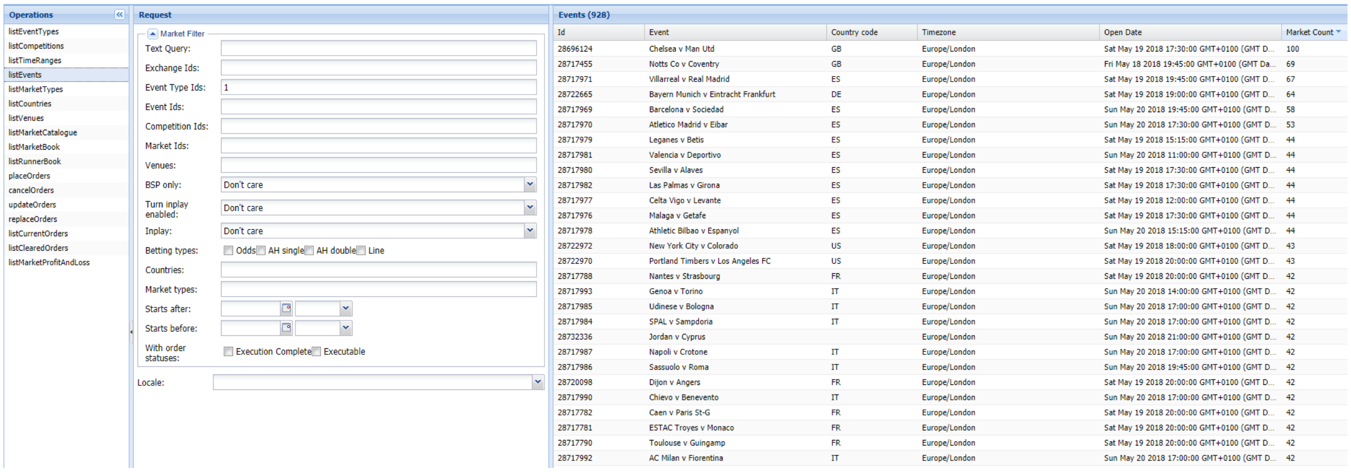


Text Query:	
Exchange Ids:	
Event Type Ids:	
Event Ids:	
Competition Ids:	
Market Ids:	
Venues:	
BSP only:	Don't care
Turn inplay enabled:	Don't care
Inplay:	Don't care
Betting types:	□ Odds □ AH single □ AH double □ Line
Countries:	
Market types:	
Starts after:	
Starts before:	
With order statuses:	□ Execution Complete □ Executable
Locale:	

EventTypes (29)

Id	EventType	Market Count
1	Soccer	18004
2	Tennis	1715
7	Horse Racing	1332
4339	Greyhound Racing	592
61420	Australian Rules	151
1477	Rugby League	135
7522	Basketball	116
5	Rugby Union	107
4	Cricket	93
26420387	Mixed Martial Arts	85
2378961	Politics	85
27454571	Esports	69
3	Golf	54
6	Boxing	53
2152880	Gaelic Games	49
468328	Handball	35
7511	Baseball	27
998917	Volleyball	26
6423	American Football	20
6231	Financial Bets	19
8	Motor Sport	15
11	Cycling	13
606611	Netball	7
7524	Ice Hockey	6
10	Special Bets	4
72382	Pool	4
3503	Darts	1
6422	Snooker	1
3988	Athletics	1

2. Request all events for EventTypeId 1 (Soccer)

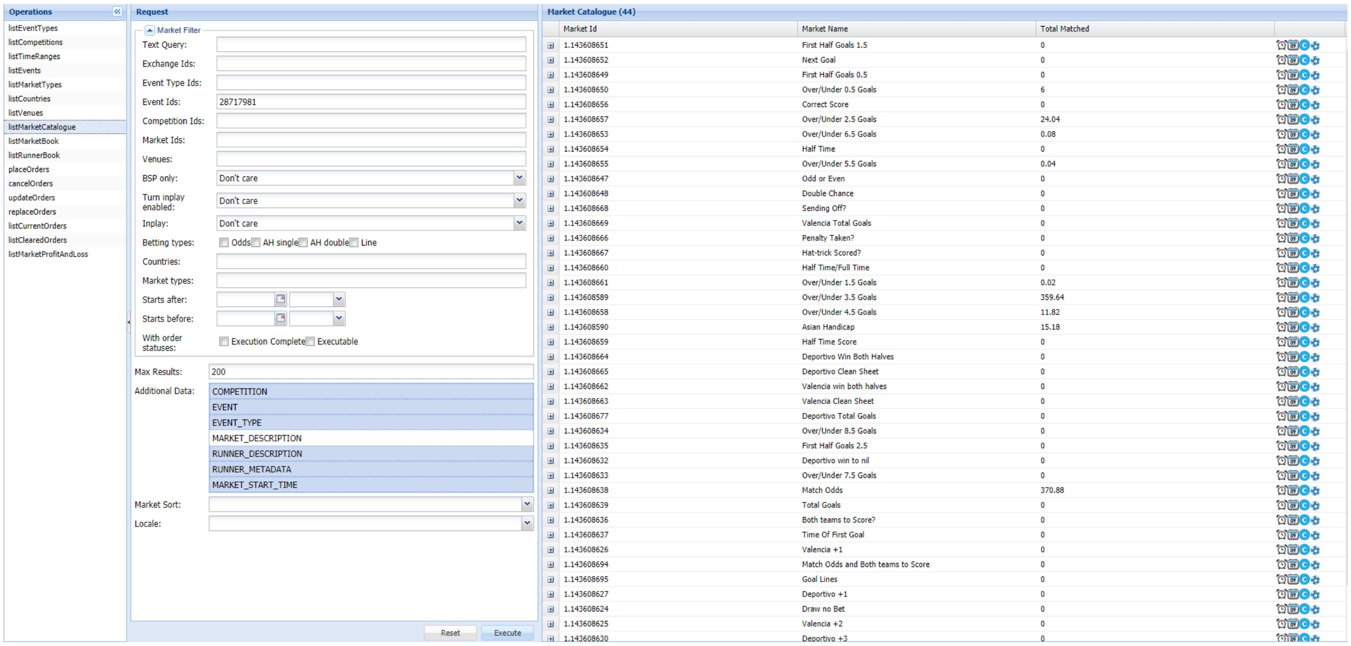


Text Query:	
Exchange Ids:	
Event Type Ids:	1
Event Ids:	
Competition Ids:	
Market Ids:	
Venues:	
BSP only:	Don't care
Turn inplay enabled:	Don't care
Inplay:	Don't care
Betting types:	□ Odds □ AH single □ AH double □ Line
Countries:	
Market types:	
Starts after:	
Starts before:	
With order statuses:	□ Execution Complete □ Executable
Locale:	

Events (928)

Id	Event	Country code	Timezone	Open Date	Market Count
28696124	Chelsea v Man Utd	GB	Europe/London	Sat May 19 2018 17:30:00 GMT+0100 (GMT D...	100
28717455	Notts Co v Coventry	GB	Europe/London	Fri May 18 2018 19:45:00 GMT+0100 (GMT Da...	69
28717971	Villarreal v Real Madrid	ES	Europe/London	Sat May 19 2018 19:45:00 GMT+0100 (GMT D...	67
28722665	Bayern Munich v Eintracht Frankfurt	DE	Europe/London	Sat May 19 2018 19:00:00 GMT+0100 (GMT D...	64
28717969	Barcelona v Sociedad	ES	Europe/London	Sun May 20 2018 19:45:00 GMT+0100 (GMT D...	58
28717970	Atletico Madrid v Eibar	ES	Europe/London	Sun May 20 2018 17:30:00 GMT+0100 (GMT D...	53
28717979	Leganes v Betis	ES	Europe/London	Sat May 19 2018 15:15:00 GMT+0100 (GMT D...	44
28717981	Valencia v Deportivo	ES	Europe/London	Sun May 20 2018 11:00:00 GMT+0100 (GMT D...	44
28717980	Sevilla v Alaves	ES	Europe/London	Sat May 19 2018 17:30:00 GMT+0100 (GMT D...	44
28717982	Las Palmas v Girona	ES	Europe/London	Sat May 19 2018 17:30:00 GMT+0100 (GMT D...	44
28717977	Celta Vigo v Levante	ES	Europe/London	Sat May 19 2018 12:00:00 GMT+0100 (GMT D...	44
28717976	Malaga v Getafe	ES	Europe/London	Sat May 19 2018 17:30:00 GMT+0100 (GMT D...	44
28717978	Athletic Bilbao v Espanyol	ES	Europe/London	Sun May 20 2018 15:15:00 GMT+0100 (GMT D...	44
28722972	New York City v Colorado	US	Europe/London	Sat May 19 2018 18:00:00 GMT+0100 (GMT D...	43
28722970	Portland Timbers v Los Angeles FC	US	Europe/London	Sat May 19 2018 20:00:00 GMT+0100 (GMT D...	43
28717788	Nantes v Strasbourg	FR	Europe/London	Sat May 19 2018 20:00:00 GMT+0100 (GMT D...	42
28717993	Genoa v Torino	IT	Europe/London	Sun May 20 2018 14:00:00 GMT+0100 (GMT D...	42
28717985	Udinese v Bologna	IT	Europe/London	Sun May 20 2018 17:00:00 GMT+0100 (GMT D...	42
28717984	SPAL v Sampdoria	IT	Europe/London	Sun May 20 2018 17:00:00 GMT+0100 (GMT D...	42
28732336	Jordan v Cyprus		Europe/London	Sun May 20 2018 21:00:00 GMT+0100 (GMT D...	42
28717987	Napoli v Crotone	IT	Europe/London	Sun May 20 2018 17:00:00 GMT+0100 (GMT D...	42
28717986	Sassuolo v Roma	IT	Europe/London	Sun May 20 2018 19:45:00 GMT+0100 (GMT D...	42
28720098	Dijon v Angers	FR	Europe/London	Sat May 19 2018 20:00:00 GMT+0100 (GMT D...	42
28717990	Chievo v Benevento	IT	Europe/London	Sun May 20 2018 17:00:00 GMT+0100 (GMT D...	42
28717782	Caen v Paris St-G	FR	Europe/London	Sat May 19 2018 20:00:00 GMT+0100 (GMT D...	42
28717781	ESTAC Troyes v Monaco	FR	Europe/London	Sat May 19 2018 20:00:00 GMT+0100 (GMT D...	42
28717790	Toulouse v Guingamp	FR	Europe/London	Sat May 19 2018 20:00:00 GMT+0100 (GMT D...	42
28717992	AC Milan v Fiorentina	IT	Europe/London	Sun May 20 2018 17:00:00 GMT+0100 (GMT D...	42

3. Request all markets for a specific eventId using listMarketCatalogue

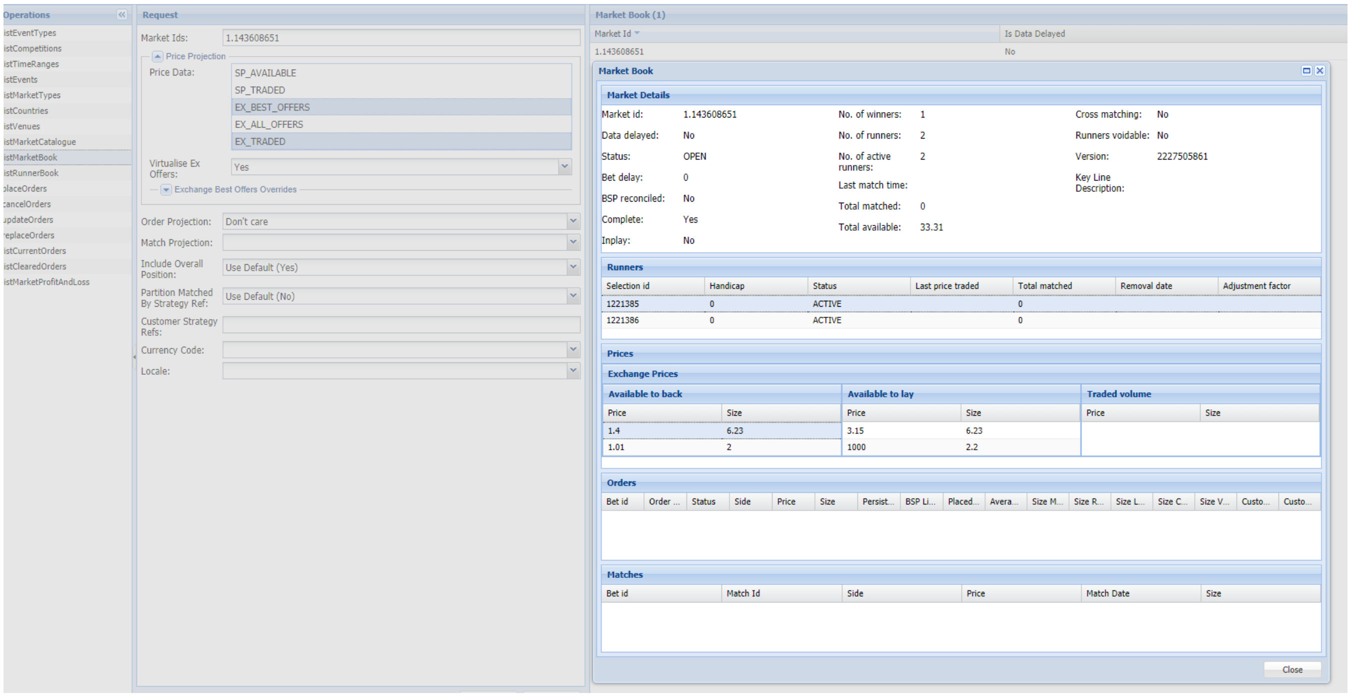


Text Query:	
Exchange Ids:	
Event Type Ids:	
Event Ids:	28717981
Competition Ids:	
Market Ids:	
Venues:	
BSP only:	Don't care
Turn inplay enabled:	Don't care
Inplay:	Don't care
Betting types:	□ Odds □ AH single □ AH double □ Line
Countries:	
Market types:	
Starts after:	
Starts before:	
With order statuses:	□ Execution Complete □ Executable
Max Results:	200
Additional Data:	COMPETITION EVENT EVENT_TYPE MARKET_DESCRIPTION RUNNER_DESCRIPTION RUNNER_METADATA MARKET_START_TIME
Market Sort:	
Locale:	

Market Catalogue (44)

Market Id	Market Name	Total Matched
1.143608651	First Half Goals 1.5	0
1.143608652	Next Goal	0
1.143608649	First Half Goals 0.5	0
1.143608650	Over/Under 0.5 Goals	6
1.143608656	Correct Score	0
1.143608657	Over/Under 2.5 Goals	24.04
1.143608653	Over/Under 6.5 Goals	0.08
1.143608654	Half Time	0
1.143608655	Over/Under 5.5 Goals	0.04
1.143608647	Odd or Even	0
1.143608648	Double Chance	0
1.143608668	Sending Off?	0
1.143608669	Valencia Total Goals	0
1.143608666	Penalty Taken?	0
1.143608667	Hat-trick Scored?	0
1.143608660	Half Time/Full Time	0
1.143608661	Over/Under 1.5 Goals	0.02
1.143608589	Over/Under 3.5 Goals	359.64
1.143608658	Over/Under 4.5 Goals	11.82
1.143608590	Asian Handicap	15.18
1.143608659	Half Time Score	0
1.143608664	Deportivo Win Both Halves	0
1.143608665	Deportivo Clean Sheet	0
1.143608662	Valencia win both halves	0
1.143608663	Valencia Clean Sheet	0
1.143608677	Deportivo Total Goals	0
1.143608634	Over/Under 8.5 Goals	0
1.143608635	First Half Goals 2.5	0
1.143608632	Deportivo win to nil	0
1.143608633	Over/Under 7.5 Goals	0
1.143608638	Match Odds	370.88
1.143608639	Total Goals	0
1.143608636	Both teams to Score?	0
1.143608637	Time Of First Goal	0
1.143608626	Valencia +1	0
1.143608694	Match Odds and Both teams to Score	0
1.143608695	Goal Lines	0
1.143608627	Deportivo +1	0
1.143608624	Draw no Bet	0
1.143608625	Valencia +2	0
1.143608630	Deportivo +3	0

4. Request prices for a specific marketId (EX_BEST_OFFERS & EX_TRADED) using listMarketBook - Please note: click on the '?' under Market Book on the right hand side to reveal the Market Details pop up.



Market Ids:	1.143608651
Price Data:	SP_AVAILABLE SP_TRADED EX_BEST_OFFERS EX_ALL_OFFERS EX_TRADED
Virtualise Ex Offers:	Yes
Order Projection:	Don't care
Match Projection:	
Include Overall Position:	Use Default (Yes)
Partition Matched By Strategy Ref:	Use Default (No)
Customer Strategy Refs:	
Currency Code:	
Locale:	

Market Book (1)

Market Id	Is Data Delayed
1.143608651	No

Market Book

Market Details

Market id:	1.143608651	No. of winners:	1	Cross matching:	No
Data delayed:	No	No. of runners:	2	Runners voidable:	No
Status:	OPEN	No. of active runners:	2	Version:	2227505861
Bet delay:	0	Last match time:		Key Line Description:	
BSP reconciled:	No	Total matched:	0		
Complete:	Yes	Total available:	33.31		
Inplay:	No				

Runners

Selection id	Handicap	Status	Last price traded	Total matched	Removal date	Adjustment factor
1221385	0	ACTIVE		0		
1221386	0	ACTIVE		0		

Prices

Exchange Prices

Available to back		Available to lay		Traded volume	
Price	Size	Price	Size	Price	Size
1.4	6.23	3.15	6.23		
1.01	2	1000	2.2		

Orders

Bet id	Order ...	Status	Side	Price	Size	Persist...	BSP Li...	Placed...	Avera...	Size M...	Size R...	Size L...	Size C...	Size V...	Custo...	Custo...

Matches

Bet id	Match Id	Side	Price	Match Date	Size

Please note

The Demo Tools are provided as-is, and is intended solely as a testing resource.

The demo tools interact directly with the Betfair API production system and not a testbed/sandbox environment.

Market Data Request Limits

What are Market Data Request Limits?

Although you can request multiple markets from listMarketBook, listRunnerBook, listMarketCatalogue and listMarketProfitandLoss, there are limits on the amount of data requested in **one request.**

sum(Weight) * **number market ids** must not exceed 200 points per requests

The following tables explain the "weighting" of data for each **MarketProjection** or **PriceProjection**. If you exceed the maximum weighting of 200 points, the API will return a **TOO_MUCH_DATA** error.

listMarketCatalogue

MarketProjection	Weight
MARKET_DESCRIPTION	1
RUNNER_DESCRIPTION	0
EVENT	0
EVENT_TYPE	0
COMPETITION	0
RUNNER_METADATA	1
MARKET_START_TIME	0

listMarketBook/listRunnerBook

PriceProjection	Weight
Null (No PriceProjection set)	2
SP_AVAILABLE	3
SP_TRADED	7
EX_BEST_OFFERS	5
EX_ALL_OFFERS	17
EX_TRADED	17

Please note: specific combinations of priceProjections will carry different weights that aren't the sum of their individual weights. Please see a summary of these below:

PriceProjection	Weight
EX_BEST_OFFERS + EX_TRADED	20
EX_ALL_OFFERS + EX_TRADED	32

If **exBestOffersOverrides** is used the weight is is calculated by **weight * (requestedDepth/3)**.

listMarketProfitandLoss

PriceProjection	Weight
Not applicable	4

Additional Information

Understanding Market Navigation

Recreating Website Navigation?

Please note: if you want to recreate the exact navigation hierarchy used by the Betfair website, please use the Navigation Data for Applications service.

In the Betfair Exchange API, navigating markets uses the concepts of faceted search. You've probably seen faceted search used on sites like eBay or Amazon where you pick a category, for example "Shoes" and then you see a list of shoes. Then, you can narrow your search by facets. For example, by colour or price. In a similar way, the new Sports API lets you find markets you're interested in and then get data back about a facet of those markets.

The way to think of the API Navigation operations is to imagine a single table that lists every market, along with various metadata about the market. The columns of the table are the Facets (and the Nav Operations in API return some of them in combination):

As you can see, the table has four markets along with some of the metadata associated with that market. Of course, the actual table would have tens of thousands of markets and more columns.

Along the top, you can see three of the navigation operations. Each of those operations takes a "MarketFilter". The MarketFilter filters the table of markets and selects the ones that match. The MarketFilter can contain things like "going in play" or "eventId 7", etc. So, if you called "listCompetitions" and passed in a filter that looked like:

Code:

```
{ "filter": {"turnsInPlay" : true }}
```

The selected markets would look like:

In the example, the 1001, 2355, and the 5621 market are the ones that match the filter. As you called "listCompetitions" with that filter, then the data returned is the columns containing competition information:

In this example, you'd get a list of competitions like so:

Code:

```
{"competitionId" : "23", "competitionName" : "World Cup 2013"}
```

Notice that the competition "Final 2013" was not returned. This is because the market was not selected by the MarketFilter. Also, notice that although the market 2355 was selected by the MarketFilter, it is not associated with a competition, so no competition id and name were returned for that market.

As a further example, suppose that you used the same MarketFilter as before while calling listEvents. In that case, the data returned is the columns containing eventId and eventName for the markets that match the filter (i.e., turnsInPlay = True):

In this example, you'd get a list of events like so:

Code:

```
[{"eventId" : "24", "eventName" : "Arsenal Vs. Reading"}, {"eventId" : "124", "eventName" : "Ascot 16th
September"}, {"eventId" : "23", "eventName" : "Cheltenham 17th September"}]
```

Again, notice that the event "Celtics vs Nicks" was not returned because it was not selected by the MarketFilter.

One further example. Suppose you were only interested in Line markets and you want to find all of the Sports (EventTypes) that contain Line markets. You would create a MarketFilter like this:

Code:

```
{ "filter": {"MarketBettingType" : "LINE" }}
```

And call listEventTypes with that filter. The markets selected by that MarketFilter would be:

and the data returned would be:

Code:

```
{"eventTypeId" : "4", "EventTypeName" : "Basketball"}
```

Using the set of navigation operations, you can easily create a menu tree of markets in whatever hierarchy you want. You can can listEventTypes to get the Sports as the top of your menu. Or, you could call "listCountries" and use markets in a country as the root of the tree.

There is also listMarketCatalogue so if you didn't care about creating a visual navigation, you could simply call listMarketCatalogue with a MarketFilter defining the markets you're interested in and get back information about those markets.

Betfair Price Increments

Below is a list of price increments per price 'group'. Placing a bet outside of these increments will result in an INVALID_ODDS error.

The Betfair Price increment used for a market is defined by the **PriceLadderType** returned within the Market Description (listMarketCatalogue / Market Definition (Exchange Stream API).

CLASSIC

Price	Increment
1.01 2	0.01
2 3	0.02
3 4	0.05
4 6	0.1
6 10	0.2
10 20	0.5
20 30	1
30 50	2
50 100	5
100 1000	10

FINEST

Price	Increment
1.01 1000	0.01

LINE_RANGE

Price	Increment (Interval)
0.1 1000	Specified by Market Line Range Info

Common Error Codes

Meaning	FaultCode	Client /Server	Associated HTTP Transport Response	Comments

			Code	
DSC -0008	JSONDeserialisationParseFailure	Client	400	
DSC -0009	ClassConversionFailure	Client	400	Invalid format for parameter, for example passing a string where a number was expected. Can also happen when a value is passed that does not match any valid enum.
DSC -0015	SecurityException	Client	403	Credentials supplied in request were invalid
DSC -0018	MandatoryNotDefined	Client	400	A parameter marked as mandatory was not provided
DSC -0019	Timeout	Server	504	The request has timed out
DSC -0021	NoSuchOperation	Client	404	The operation specified does not exist
DSC -0023	NoSuchService	Client	404	
DSC -0024	RescriptDeserialisationFailure	Client	400	Exception during deserialization of RESCRIPT request
DSC -0034	UnknownCaller	Client	400	A valid and active App Key hasn't been provided in the request. Please check that your App Key is active. Please see Application Keys for further information regarding App Keys.
DSC -0035	UnrecognisedCredentials	Client	400	
DSC -0036	InvalidCredentials	Client	400	
DSC -0037	SubscriptionRequired	Client	403	The user is not subscribed to the App Key provided
DSC -0038	OperationForbidden	Client	403	The App Key sent with the request is not permitted to access the operation

Currency Parameters

Guide to available currencies and minimum bet sizes.

Currency name	Symbol	Currency Code	Min Bet Size	Min Deposit Size	Minimum BSP Liability	Minimum Bet Payout
UK Sterling	£	GBP	2	10	10	10
Euro	€	EUR	2	15	20	20
US Dollar	US$	USD	3	15	20	20
Hong Kong Dollars	HK$	HKD	25	150	125	125
Australian Dollar	AUD	AUD	5	30	30	30
Canadian Dollar	CAD	CAD	6	25	30	30
Danish Kroner	DKK	DKK	30	150	150	150
Norwegian Kronor	NOK	NOK	30	150	150	150
Swedish Krona	SEK	SEK	30	150	150	150
Singapore Dollar	SGD	SGD	6	30	30	30
Romanian Leu	RON	RON	10	25 (Paysafe) 40 (Card)	50	50
Brazilian Real	R$	BRL	10	40	50	50
Mexican Peso	MXN	MXN	60	50	300	300
Nuevo Sol	PEN	PEN	10	50	50	50
Hungary Forint	HUF	HUF	800	25	4000	4000
Iceland Kron	ISK	ISK	350	25	1750	1750
New Zealand Dollars	NZD	NZD	2	25	10	10
Argentine Peso	ARS	ARS	100	50	500	500
Lari	GEL	GEL	10	50	50	50

Locale Specification

The locale specification determines the language returned for names of sports and markets. It is an optional parameter you can specify when you want to retrieve names in a language that differs from the language specified for the account. For example, if the account language is specified as English, you can use the locale parameter to retrieve non-English sport or market names.

The language code is based on the ISO 639-1 standard which defines two-letter codes, such as "en" and "fr".

The follow languages are available, but please be aware not all markets in all languages are fully translated:

Language	Locale Code
English	en
Danish	da
Swedish	sv
German	de
Italian	it
Greek	el
Spanish	es
Turkish	tr
Korean	ko
Czech	cs
Bulgarian	bg
Russian	ru
French	fr
Thai	th

Racecourse Abbreviations

Racecourse abbreviations lists for Horse Racing and Greyhounds is available via the spreadsheet below:

Horse Racing & ...th sept2019.xls

Runner Metadata Description

The RUNNER_METADATA returned by **listMarketCatalogue** for **Horse Racing** (when available) is described in the table below.

Parameter	Description	Example

WEIGHT_UNITS	The unit of weight used.	pounds
ADJUSTED_RATING	Adjusted ratings are race-specific ratings which reflect weights allocated in the race and, in some circumstances, the age of the horse. Collectively they represent the chance each runner has on form. https://www.timeform.com/Racing/Articles/How_the_ratings_for_a_race_are_calculated Please note: this data is only returned for those with a Premium Timeform subscription	79
DAM_YEAR_BORN	The year the horse’s mother's birth	1997
DAYS_SINCE_LAST_RUN	The number of days since the horse last ran	66
WEARING	Any extra equipment the horse is wearing	tongue strap
DAMSIRE_YEAR_BORN	The year in which the horse's grandfather was born on its mothers side	1988
SIRE_BRED	The country were the horse's father was bred	IRE
TRAINER_NAME	The name of the horse's trainer	Fergal O'Brien
STALL_DRAW	The stall number the horse is starting from	10
SEX_TYPE	The sex of the horse	f
OWNER_NAME	The owner of the horse	Mr M. C. Fahy
SIRE_NAME	The name of the horse's father	Revoque
FORECASTPRICE_NUMERATOR	The forecast price numerator	13
FORECASTPRICE_DENOMINATOR	The forecast price denominator	8
JOCKEY_CLAIM	The reduction in the weight that the horse carries for a particular jockey were applicable.	5
WEIGHT_VALUE	The weight of the horse	163
DAM_NAME	The name of the horse's mother	Rare Gesture
AGE	The age of the horse	7
COLOUR_TYPE	The colour of the horse	b
DAMSIRE_BRED	The country were the horse's grandfather was born	IRE
DAMSIRE_NAME	The name of the horse's grandfather	Shalford
SIRE_YEAR_BORN	The year the horse's father was born	1994
OFFICIAL_RATING	The horses official rating	97
FORM	The horses recent form	212246
BRED	The country in which the horse was born	IRE
runnerId	The runnerId for the horse	62434983
JOCKEY_NAME	The name of the jockey. **Please note**: This field will contain '**Reserve**' in the event that the horse has been entered into the market as a reserve runner. Any reserve runners will be withdrawn from the market once it has been confirmed that they will not run.	Paddy Brennan
DAM_BRED	The country where the horse's mother was born	IRE
COLOURS_	The textual description of the jockey silk	Royal blue

DESCRIPTION		and black check, white sleeves and cap
COLOURS_FILENAME	A relative URL to an image file corresponding to the jockey silk. You must add the value of this field to the base URL: https://content-cache.cdnbf.net/feeds_images/Horses/SilkColours/ **Please note** - silk cloth images aren't provided for US Racing. The saddlecloth images used for US racing can be view via https://sn4.cdnbf.net/exchange/plus/images/app/common/assets/images/saddlecloths-sprite_2608_.gif	c20170823trots_BATHURST/1109899.png
CLOTH_NUMBER	The number on the saddle-cloth	5
CLOTH_NUMBER ALPHA	The number on the saddle-cloth. For US Racing were the runner is paired, this field will display the cloth number of the paired runner e.g. "1A"	1A

Terms of use: Please refer to http://form.timeform.betfair.com/termsofuse regarding the above data.

Time Zones & Time Format

All times are returned in GMT and are in ISO 8601 (http://en.wikipedia.org/wiki/ISO_8601) format. They can be converted to your local timezone using the timezone field returned by the getAccountDetails operation or into the local market timezone using the timezone returned for the event by listMarketCatalogue

To synchronize with the Betfair server time we recommend that you use the pool of NTP servers listed via http://www.pool.ntp.org/zone/europe

The following table lists the time zones returned be the API along with their meaning.

Location	Abbreviation	Notes
Africa/Johannesburg	RSA	
America/Costa_Rica	SJMT	
America/Indiana/Indianapolis	IEST	North America Indiana East
America/Santiago	SMT	
Asia/Bangkok	THAI	
Asia/Calcutta	INT	
Asia/Dubai	UAE	
Australia/Adelaide	ACST	
Australia/Darwin	ANST	
Australia/Perth	AWST	
Australia/Queensland	AQST	
Australia/Sydney	AEST	
Brazil/East	BRT	
Brazil/West	AMT	
CET	CET	Central European Time
EET	EET	Eastern European Time
Etc/GMT-5	PKT	
Europe/London	UKT	
Europe/Moscow	MSK	
GMT/UTC	GMT/UTC	Greenwich Mean Time/Coordinated Universal Time
Hongkong	HK	
Jamaica	KMT	

Japan	JPT	
NZ	NZT	New Zealand
US/Alaska	AKST	
US/Arizona	AST	
US/Central	CST	
US/Eastern	EST	
US/Hawaii	HST	
US/Mountain	MST	
US/Pacific	PST	

Virtual Bets

The Betfair Exchange tries to match bets against the bets placed by other users as well as equivalent bets where possible.

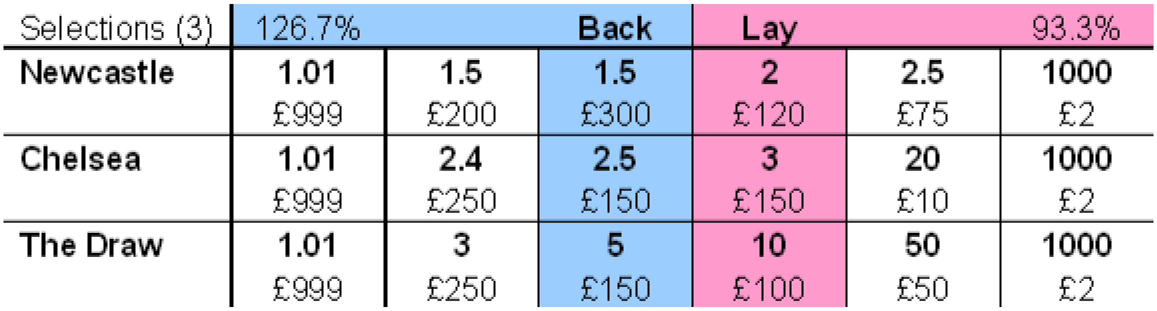
For example, if a customer backs Player A in a tennis match at odds of 2.0, the exchange matches any lay bet at odds of 2.0 or better on Player A. In addition, the Exchange also calculates all possible matches that are equivalent to a lay bet on Player A. So, if a customer backs Player B at odds of 2.0 or better, the Exchange matches it with the back bet on Player A. The Betfair Exchange uses a 'cross matching' algorithm to display the best possible prices (bets) available by taking into account the back and lay offers (unmatched bets) across all selections.

You can return virtual bets in the response when using the Exchange API by including the virtualise":"true" in the listMarketBook request e.g. [{"jsonrpc": "2.0","method": "SportsAPING/v1.0/listMarketBook", "params": {"marketIds":["1.114101556"],"priceProjection":{"priceData":["EX_BEST_OFFERS"],"virtualise":"true"}}, "id": 1}]

You should subscribe to the EX_BEST_OFFERS_DISP MarketDataFlter if using the **Exchange Stream API**

One of the easiest ways to understand how we generate virtual bets for cross matching is to work through a couple of examples.

Consider the following market and what would happen if we placed a very large back bet at 1.01 on The Draw:



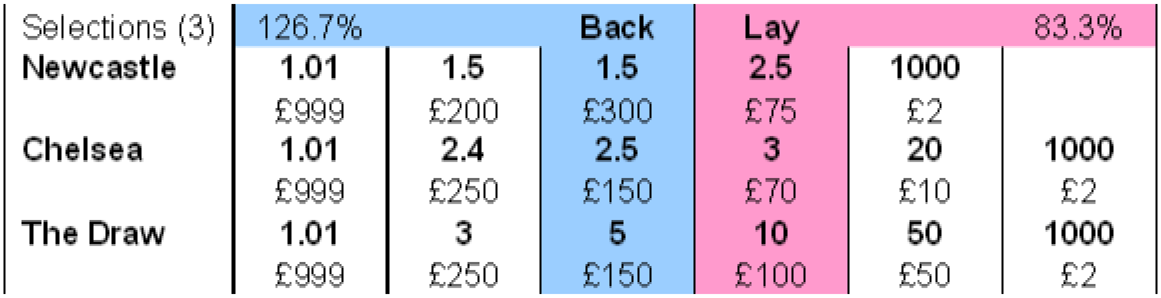
Selections (3)	126.7%		Back	Lay		93.3%
Newcastle	1.01 £999	1.5 £200	1.5 £300	2 £120	2.5 £75	1000 £2
Chelsea	1.01 £999	2.4 £250	2.5 £150	3 £150	20 £10	1000 £2
The Draw	1.01 £999	3 £250	5 £150	10 £100	50 £50	1000 £2

Without cross matching, this bet would be matched in three portions;

1) £150 at 5.0,
2) £250 at 3.0,
and £999 at 1.01 with anything remaining being left unmatched.

With cross matching we can do better.

We have a back bet on Newcastle for £120 at 2.0 and a back bet on Chelsea for £150 at 3.0 (shown in pink on the available to lay side of the market).



Selections (3)	126.7%		Back	Lay		83.3%
Newcastle	1.01 £999	1.5 £200	1.5 £300	2.5 £75	1000 £2	
Chelsea	1.01 £999	2.4 £250	2.5 £150	3 £70	20 £10	1000 £2
The Draw	1.01 £999	3 £250	5 £150	10 £100	50 £50	1000 £2

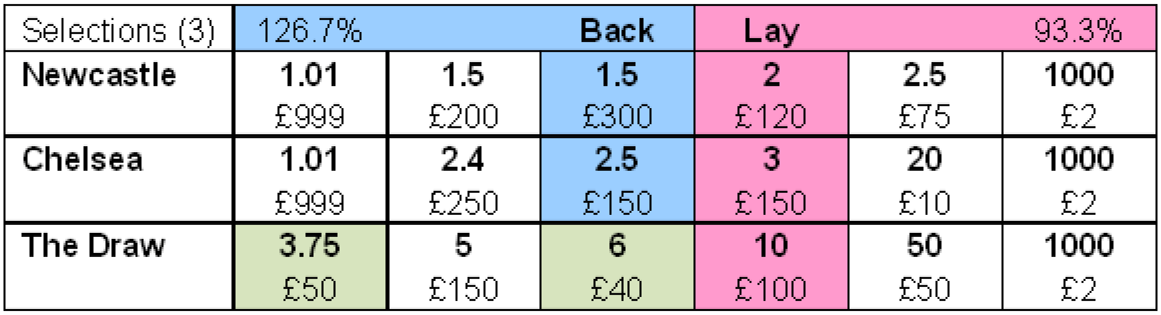
These two bets can be matched against a back bet on The Draw at a price of 6.0, since 2.0, 3.0, and 6.0 form a 100% book. To ensure that the book is balanced, we choose the stakes to be inversely proportional to the prices.

This means that we take the full £120 at 2.0 on Newcastle, only £80 at 3.0 on Chelsea, and £40 at 6.0 on The Draw, which is the first virtual bet

We now have a back bet on Newcastle for £75 at 2.5 and a back bet on Chelsea for £70 at 3.0 (again shown in pink on the available to lay side of the market).

These two bets can be matched against a back bet on The Draw at a price of 3.75, since 2.5, 3.0, and 3.75 also form a 100% book.

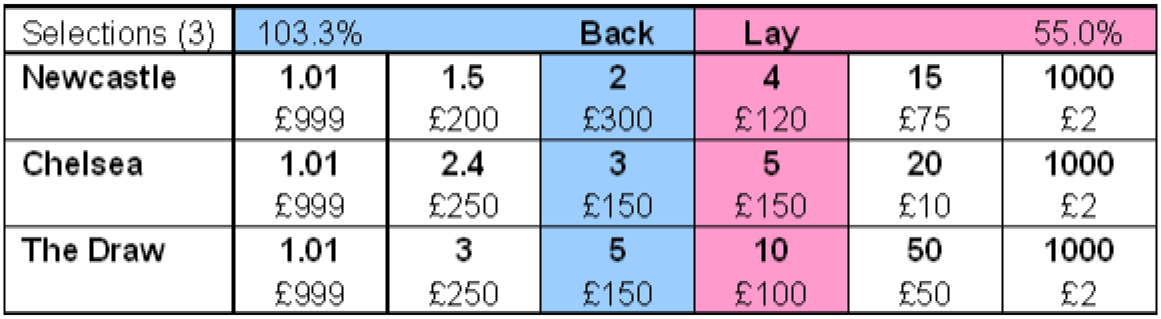
Balancing the stakes means that we need to take the full £75 at 2.5 on Newcastle, only £62.50 at 3.0 on Chelsea, and £50 at 3.75 on The Draw, which is the second virtual bet. Since 3.75 is less than 5.0, the £150 at 5.0 would be matched first followed by £50 at 3.75. If we continued this process we would get further matching at 1.50 and 1.05, but for the purposes of displaying the market view we have the best 3 prices for the available to back bets on The Draw, and so we can stop calculating the virtual bets. The virtual bets are just the bets that would have been matched had we received a sufficiently large back bet at 1.01; in this example, £40 at 6.0 and £50 at 3.75. **We take these virtual bets and merge them with the existing bets on the market to generate the following market view (with the virtual bets shown in green)**



Selections (3)	126.7%		Back	Lay		93.3%
Newcastle	1.01 £999	1.5 £200	1.5 £300	2 £120	2.5 £75	1000 £2
Chelsea	1.01 £999	2.4 £250	2.5 £150	3 £150	20 £10	1000 £2
The Draw	3.75 £50	5 £150	6 £40	10 £100	50 £50	1000 £2

The process is repeated to obtain the virtual lay bets (available to back bets) for Newcastle and Chelsea.

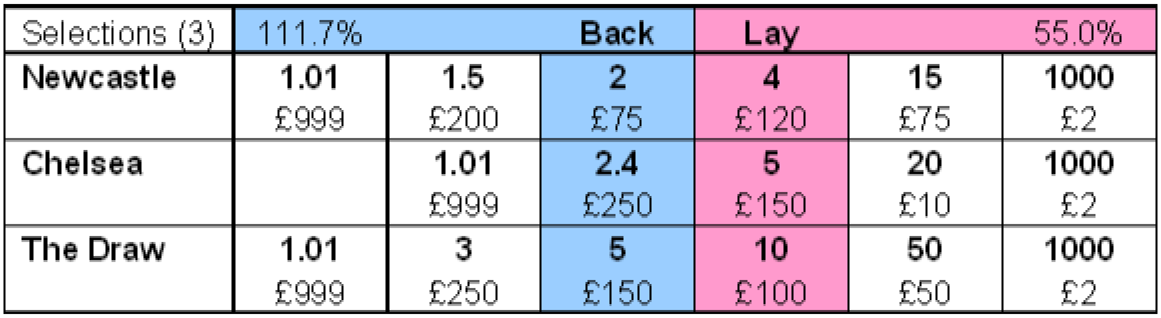
Here we have a slightly different market (as before, chosen to make the numbers nice) and consider what would happen if we placed a very large lay bet at 1000 on The Draw.



Selections (3)	103.3%		Back	Lay		55.0%
Newcastle	1.01 £999	1.5 £200	2 £300	4 £120	15 £75	1000 £2
Chelsea	1.01 £999	2.4 £250	3 £150	5 £150	20 £10	1000 £2
The Draw	1.01 £999	3 £250	5 £150	10 £100	50 £50	1000 £2

Without cross matching, this bet would be matched in three portions; 1) £100 at 10.0, 2) £50 at 50.0, and £2 at 1000 with anything remaining being left unmatched. With cross matching we can do better. We have a lay bet on Newcastle for £300 at 2.0 and a lay bet on Chelsea for £150 at 3.0 (shown in blue on the available to back side of the market). These two bets can be matched against a lay bet on The Draw at a price of 6.0, since 2.0, 3.0, and 6.0 form a 100% book. To ensure that the book is balanced, we choose the stakes to be inversely proportional to the prices. This means that we take only £225 at 2.0 on Newcastle, the full £150 at 3.0 on Chelsea, and £75 at 6.0 on The Draw, which is the first virtual bet.

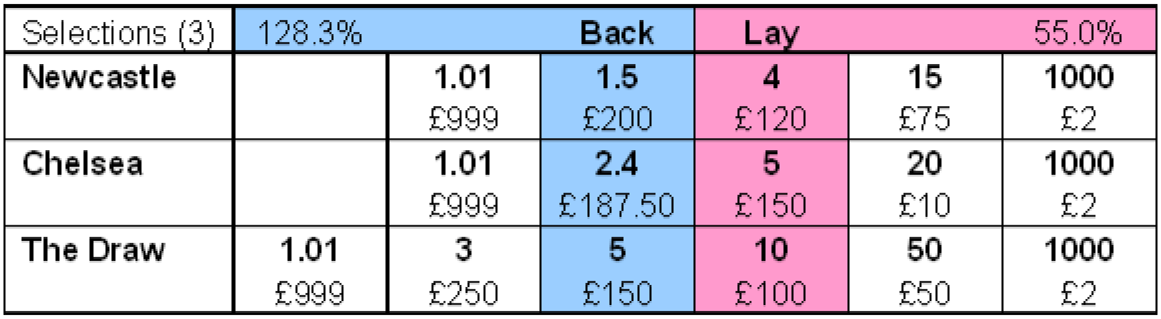
Assuming these bets got matched, the market would look like this:



Selections (3)	111.7%		Back	Lay		55.0%
Newcastle	1.01 £999	1.5 £200	2 £75	4 £120	15 £75	1000 £2
Chelsea		1.01 £999	2.4 £250	5 £150	20 £10	1000 £2
The Draw	1.01 £999	3 £250	5 £150	10 £100	50 £50	1000 £2

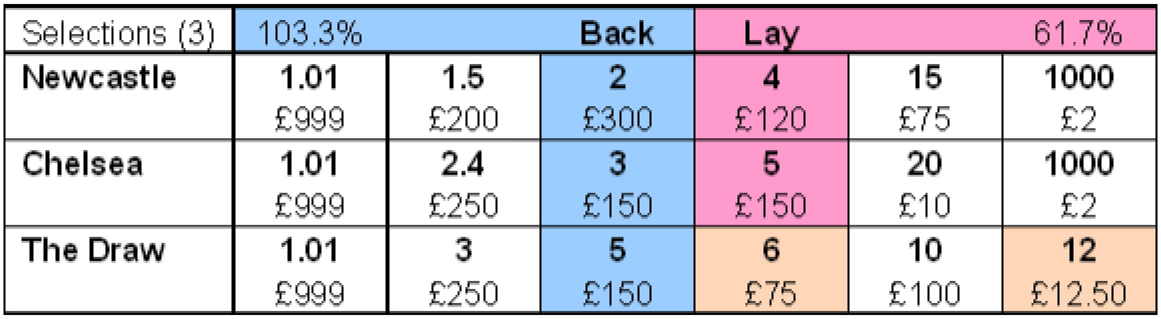
We now have a lay bet on Newcastle for £75 at 2.0 and a lay bet on Chelsea for £250 at 2.4 (again shown in blue on the available to back side of the market). These two bets can be matched against a lay bet on The Draw at a price of 12.0, since 2.0, 2.4, and 12.0 also form a 100% book. Balancing the stakes means that we need to take the full £75 at 2.0 on Newcastle, only £62.50 at 2.4 on Chelsea, and £12.50 at 12.0 on The Draw, which is the second virtual bet.

This leaves the following market:



Selections (3)	128.3%		Back	Lay		55.0%
Newcastle		1.01 £999	1.5 £200	4 £120	15 £75	1000 £2
Chelsea		1.01 £999	2.4 £187.50	5 £150	20 £10	1000 £2
The Draw	1.01 £999	3 £250	5 £150	10 £100	50 £50	1000 £2

This time we can't continue the process since there is no valid price for a virtual bet on The Draw that would result in a 100% book, and so we can stop calculating the virtual bets. Again, the virtual bets are just the bets that would have been matched had we received a sufficiently large lay bet at 1000; in this example, £75 at 6.0 and £12.50 at 12.0. **We take these virtual bets and merge them with the existing bets on the market to generate the following market view (with the virtual bets shown in orange):**



Selections (3)	103.3%		Back	Lay		61.7%
Newcastle	1.01 £999	1.5 £200	2 £300	4 £120	15 £75	1000 £2
Chelsea	1.01 £999	2.4 £250	3 £150	5 £150	20 £10	1000 £2
The Draw	1.01 £999	3 £250	5 £150	6 £75	10 £100	12 £12.50

Account Statement - Resettlement Lifecycle

The below demonstrates the resettlement lifecycle and how this is represented via the getAccountStatement output.

In the first file, the back bet on Landskrona was settled as winner, and so we see two entries for that bet:

- The initial debit (RESULT_NOT_APPLICABLE)
- The profit from winning the bet (RESULT)

Initial Settlement

```
{
    "jsonrpc": "2.0",
    "result": {
        "accountStatement": [
            {
                "refId": "10841763486",
                "itemDate": "2018-06-07T09:30:27.000Z",
                "amount": -0.9,
                "balance": 1000034.2,
                "itemClassData": {
                    "unknownStatementItem": "{\"avgPrice\":0.0,\"betSize\":0.0,\"betType\":\"B\",\"
betCategoryType\":\"NONE\",\"commissionRate\":\"5%\",\"eventId\":102828127,\"eventTypeId\":1,\"fullMarketName\":
\"Fixtures 03 June      / Landskrona v GAIS/ Match Odds\",\"grossBetAmount\":18.0,\"marketName\":\"Match Odds\",
\"marketType\":\"O\",\"placedDate\":\"2018-06-07T09:27:58.000Z\",\"selectionId\":0,\"selectionName\":null,\"
startDate\":\"0001-01-01T00:00:00.000Z\",\"transactionType\":\"ACCOUNT_DEBIT\",\"transactionId\":0,\"winLose\":
\"RESULT_NOT_APPLICABLE\"}"
                },
                "legacyData": {
                    "avgPrice": 0,
                    "betSize": 0,
                    "betType": "B",
                    "betCategoryType": "NONE",
                    "commissionRate": "5%",
                    "eventId": 102828127,
                    "eventTypeId": 1,
                    "fullMarketName": "Fixtures 03 June      / Landskrona v GAIS/ Match Odds",
                    "grossBetAmount": 18,
                    "marketName": "Match Odds",
                    "marketType": "O",
                    "placedDate": "2018-06-07T09:27:58.000Z",
                    "selectionId": 0,
                    "startDate": "0001-01-01T00:00:00.000Z",
                    "transactionType": "ACCOUNT_DEBIT",
                    "transactionId": 0,
                    "winLose": "RESULT_NOT_APPLICABLE"
                },
                "itemClass": "UNKNOWN"
            },
            {
                "refId": "10841763486",
                "itemDate": "2018-06-07T09:30:27.000Z",
                "amount": 18,
                "balance": 1000035.1,
                "itemClassData": {
                    "unknownStatementItem": "{\"avgPrice\":10.0,\"betSize\":2.0,\"betType\":\"B\",\"
betCategoryType\":\"E\",\"commissionRate\":null,\"eventId\":102828127,\"eventTypeId\":1,\"fullMarketName\":\"
Fixtures 03 June      / Landskrona v GAIS/ Match Odds\",\"grossBetAmount\":0.0,\"marketName\":\"Match Odds\",\"
marketType\":\"O\",\"placedDate\":\"2018-06-07T09:27:58.000Z\",\"selectionId\":130432,\"selectionName\":\"
Landskrona\",\"startDate\":\"0001-01-01T00:00:00.000Z\",\"transactionType\":\"ACCOUNT_CREDIT\",\"
transactionId\":0,\"winLose\":\"RESULT_WON\"}"
                },
                "legacyData": {
                    "avgPrice": 10,
                    "betSize": 2,
                    "betType": "B",
                    "betCategoryType": "E",
                    "eventId": 102828127,
                    "eventTypeId": 1,
                    "fullMarketName": "Fixtures 03 June      / Landskrona v GAIS/ Match Odds",
                    "grossBetAmount": 0,
                    "marketName": "Match Odds",
                    "marketType": "O",
                    "placedDate": "2018-06-07T09:27:58.000Z",
                    "selectionId": 130432,
                    "selectionName": "Landskrona",
                    "startDate": "0001-01-01T00:00:00.000Z",
                    "transactionType": "ACCOUNT_CREDIT",
                    "transactionId": 0,
                    "winLose": "RESULT_WON"
                },
                "itemClass": "UNKNOWN"
            },
            {
                "refId": "10841762387",
                "itemDate": "2018-06-07T09:14:03.000Z",
                "amount": -0.9,
                "balance": 1000051.3,
                "itemClassData": {
                    "unknownStatementItem": "{\"avgPrice\":0.0,\"betSize\":0.0,\"betType\":\"B\",\"
betCategoryType\":\"NONE\",\"commissionRate\":\"5%\",\"eventId\":102833966,\"eventTypeId\":7,\"fullMarketName\":
\"GB / Uttox  6th Jun/ 13:40 2m Hcap Hrd\",\"grossBetAmount\":18.0,\"marketName\":\"2m Hcap Hrd\",\"
marketType\":\"O\",\"placedDate\":\"2018-06-06T16:37:13.000Z\",\"selectionId\":0,\"selectionName\":null,\"
startDate\":\"2018-06-06T12:40:00.000Z\",\"transactionType\":\"ACCOUNT_DEBIT\",\"transactionId\":0,\"winLose\":
\"RESULT_NOT_APPLICABLE\"}"
                },
                "legacyData": {
                    "avgPrice": 0,
                    "betSize": 0,
                    "betType": "B",
                    "betCategoryType": "NONE",
                    "commissionRate": "5%",
                    "eventId": 102833966,
                    "eventTypeId": 7,
                    "fullMarketName": "GB / Uttox  6th Jun/ 13:40 2m Hcap Hrd",
                    "grossBetAmount": 18,
                    "marketName": "2m Hcap Hrd",
                    "marketType": "O",
                    "placedDate": "2018-06-06T16:37:13.000Z",
                    "selectionId": 0,
                    "startDate": "2018-06-06T12:40:00.000Z",
                    "transactionType": "ACCOUNT_DEBIT",
                    "transactionId": 0,
                    "winLose": "RESULT_NOT_APPLICABLE"
                },
                "itemClass": "UNKNOWN"
            },
            {
                "refId": "10841762387",
                "itemDate": "2018-06-07T09:14:03.000Z",
                "amount": 18,
                "balance": 1000052.2,
                "itemClassData": {
                    "unknownStatementItem": "{\"avgPrice\":10.0,\"betSize\":2.0,\"betType\":\"B\",\"
betCategoryType\":\"E\",\"commissionRate\":null,\"eventId\":102833966,\"eventTypeId\":7,\"fullMarketName\":\"GB
/ Uttox  6th Jun/ 13:40 2m Hcap Hrd\",\"grossBetAmount\":0.0,\"marketName\":\"2m Hcap Hrd\",\"marketType\":\"
O\",\"placedDate\":\"2018-06-06T16:37:13.000Z\",\"selectionId\":4700101,\"selectionName\":\"Catchin Time\",\"
startDate\":\"2018-06-06T12:40:00.000Z\",\"transactionType\":\"ACCOUNT_CREDIT\",\"transactionId\":0,\"winLose\":
\"RESULT_WON\"}"
                },
                "legacyData": {
                    "avgPrice": 10,
                    "betSize": 2,
                    "betType": "B",
                    "betCategoryType": "E",
                    "eventId": 102833966,
                    "eventTypeId": 7,
                    "fullMarketName": "GB / Uttox  6th Jun/ 13:40 2m Hcap Hrd",
                    "grossBetAmount": 0,
                    "marketName": "2m Hcap Hrd",
                    "marketType": "O",
                    "placedDate": "2018-06-06T16:37:13.000Z",
                    "selectionId": 4700101,
                    "selectionName": "Catchin Time",
                    "startDate": "2018-06-06T12:40:00.000Z",
                    "transactionType": "ACCOUNT_CREDIT",
                    "transactionId": 0,
                    "winLose": "RESULT_WON"
                },
                "itemClass": "UNKNOWN"
            }
        ],
        "moreAvailable": false
    }
}
```

In the second example, the market was unsettled, and we see four entries:

- The two entries described above are still present but tagged with RESULT_ERR
- An entry to revert the commission payed (tagged with COMMISSION_REVERSAL and RESULT_FIX)
- An entry to undo the payout from the winning bet (tagged with RESULT_FIX)

Market Unsettled

```
{
    "jsonrpc": "2.0",
    "result": {
        "accountStatement": [
            {
                "refId": "10841763486",
                "itemDate": "2018-06-07T13:20:10.000Z",
                "amount": 0.9,
                "balance": 1000034.2,
                "itemClassData": {
                    "unknownStatementItem": "{\"avgPrice\":0.0,\"betSize\":0.0,\"betType\":\"B\",\"
betCategoryType\":\"E\",\"commissionRate\":null,\"eventId\":102828127,\"eventTypeId\":1,\"fullMarketName\":\"
Fixtures 03 June      / Landskrona v GAIS/ Match Odds\",\"grossBetAmount\":0.0,\"marketName\":\"Match Odds\",\"
marketType\":\"O\",\"placedDate\":\"2018-06-07T13:20:10.000Z\",\"selectionId\":0,\"selectionName\":null,\"
startDate\":\"0001-01-01T00:00:00.000Z\",\"transactionType\":\"COMMISSION_REVERSAL\",\"transactionId\":0,\"
winLose\":\"RESULT_FIX\"}"
                },
                "legacyData": {
                    "avgPrice": 0,
                    "betSize": 0,
                    "betType": "B",
                    "betCategoryType": "E",
                    "eventId": 102828127,
                    "eventTypeId": 1,
                    "fullMarketName": "Fixtures 03 June      / Landskrona v GAIS/ Match Odds",
                    "grossBetAmount": 0,
                    "marketName": "Match Odds",
                    "marketType": "O",
                    "placedDate": "2018-06-07T13:20:10.000Z",
                    "selectionId": 0,
                    "startDate": "0001-01-01T00:00:00.000Z",
                    "transactionType": "COMMISSION_REVERSAL",
                    "transactionId": 0,
                    "winLose": "RESULT_FIX"
                },
                "itemClass": "UNKNOWN"
            },
            {
                "refId": "10841763486",
                "itemDate": "2018-06-07T13:20:10.000Z",
                "amount": -18,
                "balance": 1000033.3,
                "itemClassData": {
                    "unknownStatementItem": "{\"avgPrice\":0.0,\"betSize\":0.0,\"betType\":\"B\",\"
betCategoryType\":\"E\",\"commissionRate\":null,\"eventId\":102828127,\"eventTypeId\":1,\"fullMarketName\":\"
Fixtures 03 June      / Landskrona v GAIS/ Match Odds\",\"grossBetAmount\":0.0,\"marketName\":\"Match Odds\",\"
marketType\":\"O\",\"placedDate\":\"2018-06-07T13:20:10.000Z\",\"selectionId\":0,\"selectionName\":null,\"
startDate\":\"0001-01-01T00:00:00.000Z\",\"transactionType\":\"ACCOUNT_DEBIT\",\"transactionId\":0,\"winLose\":
\"RESULT_FIX\"}"
                },
                "legacyData": {
                    "avgPrice": 0,
                    "betSize": 0,
                    "betType": "B",
                    "betCategoryType": "E",
                    "eventId": 102828127,
                    "eventTypeId": 1,
                    "fullMarketName": "Fixtures 03 June      / Landskrona v GAIS/ Match Odds",
                    "grossBetAmount": 0,
                    "marketName": "Match Odds",
                    "marketType": "O",
                    "placedDate": "2018-06-07T13:20:10.000Z",
                    "selectionId": 0,
                    "startDate": "0001-01-01T00:00:00.000Z",
                    "transactionType": "ACCOUNT_DEBIT",
                    "transactionId": 0,
                    "winLose": "RESULT_FIX"
                },
                "itemClass": "UNKNOWN"
            },
            {
                "refId": "10841763486",
                "itemDate": "2018-06-07T09:30:27.000Z",
                "amount": -0.9,
                "balance": 1000034.2,
                "itemClassData": {
                    "unknownStatementItem": "{\"avgPrice\":0.0,\"betSize\":0.0,\"betType\":\"B\",\"
betCategoryType\":\"NONE\",\"commissionRate\":\"5%\",\"eventId\":102828127,\"eventTypeId\":1,\"fullMarketName\":
\"Fixtures 03 June      / Landskrona v GAIS/ Match Odds\",\"grossBetAmount\":18.0,\"marketName\":\"Match Odds\",
\"marketType\":\"O\",\"placedDate\":\"2018-06-07T09:27:58.000Z\",\"selectionId\":0,\"selectionName\":null,\"
startDate\":\"0001-01-01T00:00:00.000Z\",\"transactionType\":\"ACCOUNT_DEBIT\",\"transactionId\":0,\"winLose\":
\"RESULT_ERR\"}"
                },
                "legacyData": {
                    "avgPrice": 0,
                    "betSize": 0,
                    "betType": "B",
                    "betCategoryType": "NONE",
                    "commissionRate": "5%",
                    "eventId": 102828127,
                    "eventTypeId": 1,
                    "fullMarketName": "Fixtures 03 June      / Landskrona v GAIS/ Match Odds",
                    "grossBetAmount": 18,
                    "marketName": "Match Odds",
                    "marketType": "O",
                    "placedDate": "2018-06-07T09:27:58.000Z",
                    "selectionId": 0,
                    "startDate": "0001-01-01T00:00:00.000Z",
                    "transactionType": "ACCOUNT_DEBIT",
                    "transactionId": 0,
                    "winLose": "RESULT_ERR"
                },
                "itemClass": "UNKNOWN"
            },
            {
                "refId": "10841763486",
                "itemDate": "2018-06-07T09:30:27.000Z",
                "amount": 18,
                "balance": 1000035.1,
                "itemClassData": {
                    "unknownStatementItem": "{\"avgPrice\":10.0,\"betSize\":2.0,\"betType\":\"B\",\"
betCategoryType\":\"E\",\"commissionRate\":null,\"eventId\":102828127,\"eventTypeId\":1,\"fullMarketName\":\"
Fixtures 03 June      / Landskrona v GAIS/ Match Odds\",\"grossBetAmount\":0.0,\"marketName\":\"Match Odds\",\"
marketType\":\"O\",\"placedDate\":\"2018-06-07T09:27:58.000Z\",\"selectionId\":130432,\"selectionName\":\"
Landskrona\",\"startDate\":\"0001-01-01T00:00:00.000Z\",\"transactionType\":\"ACCOUNT_CREDIT\",\"
transactionId\":0,\"winLose\":\"RESULT_ERR\"}"
                },
                "legacyData": {
                    "avgPrice": 10,
                    "betSize": 2,
                    "betType": "B",
                    "betCategoryType": "E",
                    "eventId": 102828127,
                    "eventTypeId": 1,
                    "fullMarketName": "Fixtures 03 June      / Landskrona v GAIS/ Match Odds",
                    "grossBetAmount": 0,
                    "marketName": "Match Odds",
                    "marketType": "O",
                    "placedDate": "2018-06-07T09:27:58.000Z",
                    "selectionId": 130432,
                    "selectionName": "Landskrona",
                    "startDate": "0001-01-01T00:00:00.000Z",
                    "transactionType": "ACCOUNT_CREDIT",
                    "transactionId": 0,
                    "winLose": "RESULT_ERR"
                },
                "itemClass": "UNKNOWN"
            },
            {
                "refId": "10841762387",
                "itemDate": "2018-06-07T09:14:03.000Z",
                "amount": -0.9,
                "balance": 1000051.3,
                "itemClassData": {
                    "unknownStatementItem": "{\"avgPrice\":0.0,\"betSize\":0.0,\"betType\":\"B\",\"
betCategoryType\":\"NONE\",\"commissionRate\":\"5%\",\"eventId\":102833966,\"eventTypeId\":7,\"fullMarketName\":
\"GB / Uttox  6th Jun/ 13:40 2m Hcap Hrd\",\"grossBetAmount\":18.0,\"marketName\":\"2m Hcap Hrd\",\"
marketType\":\"O\",\"placedDate\":\"2018-06-06T16:37:13.000Z\",\"selectionId\":0,\"selectionName\":null,\"
startDate\":\"2018-06-06T12:40:00.000Z\",\"transactionType\":\"ACCOUNT_DEBIT\",\"transactionId\":0,\"winLose\":
\"RESULT_NOT_APPLICABLE\"}"
                },
                "legacyData": {
                    "avgPrice": 0,
                    "betSize": 0,
                    "betType": "B",
                    "betCategoryType": "NONE",
                    "commissionRate": "5%",
                    "eventId": 102833966,
                    "eventTypeId": 7,
                    "fullMarketName": "GB / Uttox  6th Jun/ 13:40 2m Hcap Hrd",
                    "grossBetAmount": 18,
                    "marketName": "2m Hcap Hrd",
                    "marketType": "O",
                    "placedDate": "2018-06-06T16:37:13.000Z",
                    "selectionId": 0,
                    "startDate": "2018-06-06T12:40:00.000Z",
                    "transactionType": "ACCOUNT_DEBIT",
                    "transactionId": 0,
                    "winLose": "RESULT_NOT_APPLICABLE"
                },
                "itemClass": "UNKNOWN"
            },
            {
                "refId": "10841762387",
                "itemDate": "2018-06-07T09:14:03.000Z",
                "amount": 18,
                "balance": 1000052.2,
                "itemClassData": {
                    "unknownStatementItem": "{\"avgPrice\":10.0,\"betSize\":2.0,\"betType\":\"B\",\"
betCategoryType\":\"E\",\"commissionRate\":null,\"eventId\":102833966,\"eventTypeId\":7,\"fullMarketName\":\"GB
/ Uttox  6th Jun/ 13:40 2m Hcap Hrd\",\"grossBetAmount\":0.0,\"marketName\":\"2m Hcap Hrd\",\"marketType\":\"
O\",\"placedDate\":\"2018-06-06T16:37:13.000Z\",\"selectionId\":4700101,\"selectionName\":\"Catchin Time\",\"
startDate\":\"2018-06-06T12:40:00.000Z\",\"transactionType\":\"ACCOUNT_CREDIT\",\"transactionId\":0,\"winLose\":
\"RESULT_WON\"}"
                },
                "legacyData": {
                    "avgPrice": 10,
                    "betSize": 2,
                    "betType": "B",
                    "betCategoryType": "E",
                    "eventId": 102833966,
                    "eventTypeId": 7,
                    "fullMarketName": "GB / Uttox  6th Jun/ 13:40 2m Hcap Hrd",
                    "grossBetAmount": 0,
                    "marketName": "2m Hcap Hrd",
                    "marketType": "O",
                    "placedDate": "2018-06-06T16:37:13.000Z",
                    "selectionId": 4700101,
                    "selectionName": "Catchin Time",
                    "startDate": "2018-06-06T12:40:00.000Z",
                    "transactionType": "ACCOUNT_CREDIT",
                    "transactionId": 0,
                    "winLose": "RESULT_WON"
                },
                "itemClass": "UNKNOWN"
            }
        ],
        "moreAvailable": false
    }
}
```

In the third example, we resettle the market with the back bet on Landskrona as a loser, and we see 5 entries:

- The four entries described above
- An entry describing the losing bet (RESULT_LOST)

Market Resettled

```
{
    "jsonrpc": "2.0",
    "result": {
        "accountStatement": [
            {
                "refId": "10841763486",
                "itemDate": "2018-06-07T13:29:52.000Z",
                "amount": -2,
                "balance": 1000032.2,
                "itemClassData": {
                    "unknownStatementItem": "{\"avgPrice\":10.0,\"betSize\":2.0,\"betType\":\"B\",\"
betCategoryType\":\"E\",\"commissionRate\":null,\"eventId\":102828127,\"eventTypeId\":1,\"fullMarketName\":\"
Fixtures 03 June      / Landskrona v GAIS/ Match Odds\",\"grossBetAmount\":0.0,\"marketName\":\"Match Odds\",\"
marketType\":\"O\",\"placedDate\":\"2018-06-07T09:27:58.000Z\",\"selectionId\":130432,\"selectionName\":\"
Landskrona\",\"startDate\":\"0001-01-01T00:00:00.000Z\",\"transactionType\":\"ACCOUNT_DEBIT\",\"transactionId\":
0,\"winLose\":\"RESULT_LOST\"}"
                },
                "legacyData": {
                    "avgPrice": 10,
                    "betSize": 2,
                    "betType": "B",
                    "betCategoryType": "E",
                    "eventId": 102828127,
                    "eventTypeId": 1,
                    "fullMarketName": "Fixtures 03 June      / Landskrona v GAIS/ Match Odds",
                    "grossBetAmount": 0,
                    "marketName": "Match Odds",
                    "marketType": "O",
                    "placedDate": "2018-06-07T09:27:58.000Z",
                    "selectionId": 130432,
                    "selectionName": "Landskrona",
                    "startDate": "0001-01-01T00:00:00.000Z",
                    "transactionType": "ACCOUNT_DEBIT",
                    "transactionId": 0,
                    "winLose": "RESULT_LOST"
                },
                "itemClass": "UNKNOWN"
            },
            {
                "refId": "10841763486",
                "itemDate": "2018-06-07T13:20:10.000Z",
                "amount": 0.9,
                "balance": 1000034.2,
                "itemClassData": {
                    "unknownStatementItem": "{\"avgPrice\":0.0,\"betSize\":0.0,\"betType\":\"B\",\"
betCategoryType\":\"E\",\"commissionRate\":null,\"eventId\":102828127,\"eventTypeId\":1,\"fullMarketName\":\"
Fixtures 03 June      / Landskrona v GAIS/ Match Odds\",\"grossBetAmount\":0.0,\"marketName\":\"Match Odds\",\"
marketType\":\"O\",\"placedDate\":\"2018-06-07T13:20:10.000Z\",\"selectionId\":0,\"selectionName\":null,\"
startDate\":\"0001-01-01T00:00:00.000Z\",\"transactionType\":\"COMMISSION_REVERSAL\",\"transactionId\":0,\"
winLose\":\"RESULT_FIX\"}"
                },
                "legacyData": {
                    "avgPrice": 0,
                    "betSize": 0,
                    "betType": "B",
                    "betCategoryType": "E",
                    "eventId": 102828127,
                    "eventTypeId": 1,
                    "fullMarketName": "Fixtures 03 June      / Landskrona v GAIS/ Match Odds",
                    "grossBetAmount": 0,
                    "marketName": "Match Odds",
                    "marketType": "O",
                    "placedDate": "2018-06-07T13:20:10.000Z",
                    "selectionId": 0,
                    "startDate": "0001-01-01T00:00:00.000Z",
                    "transactionType": "COMMISSION_REVERSAL",
                    "transactionId": 0,
                    "winLose": "RESULT_FIX"
                },
                "itemClass": "UNKNOWN"
            },
            {
                "refId": "10841763486",
                "itemDate": "2018-06-07T13:20:10.000Z",
                "amount": -18,
                "balance": 1000033.3,
                "itemClassData": {
                    "unknownStatementItem": "{\"avgPrice\":0.0,\"betSize\":0.0,\"betType\":\"B\",\"
betCategoryType\":\"E\",\"commissionRate\":null,\"eventId\":102828127,\"eventTypeId\":1,\"fullMarketName\":\"
Fixtures 03 June      / Landskrona v GAIS/ Match Odds\",\"grossBetAmount\":0.0,\"marketName\":\"Match Odds\",\"
marketType\":\"O\",\"placedDate\":\"2018-06-07T13:20:10.000Z\",\"selectionId\":0,\"selectionName\":null,\"
startDate\":\"0001-01-01T00:00:00.000Z\",\"transactionType\":\"ACCOUNT_DEBIT\",\"transactionId\":0,\"winLose\":
\"RESULT_FIX\"}"
                },
                "legacyData": {
                    "avgPrice": 0,
                    "betSize": 0,
                    "betType": "B",
                    "betCategoryType": "E",
                    "eventId": 102828127,
                    "eventTypeId": 1,
                    "fullMarketName": "Fixtures 03 June      / Landskrona v GAIS/ Match Odds",
                    "grossBetAmount": 0,
                    "marketName": "Match Odds",
                    "marketType": "O",
                    "placedDate": "2018-06-07T13:20:10.000Z",
                    "selectionId": 0,
                    "startDate": "0001-01-01T00:00:00.000Z",
                    "transactionType": "ACCOUNT_DEBIT",
                    "transactionId": 0,
                    "winLose": "RESULT_FIX"
                },
                "itemClass": "UNKNOWN"
            },
            {
                "refId": "10841763486",
                "itemDate": "2018-06-07T09:30:27.000Z",
                "amount": -0.9,
                "balance": 1000034.2,
                "itemClassData": {
                    "unknownStatementItem": "{\"avgPrice\":0.0,\"betSize\":0.0,\"betType\":\"B\",\"
betCategoryType\":\"NONE\",\"commissionRate\":\"5%\",\"eventId\":102828127,\"eventTypeId\":1,\"fullMarketName\":
\"Fixtures 03 June      / Landskrona v GAIS/ Match Odds\",\"grossBetAmount\":18.0,\"marketName\":\"Match Odds\",
\"marketType\":\"O\",\"placedDate\":\"2018-06-07T09:27:58.000Z\",\"selectionId\":0,\"selectionName\":null,\"
startDate\":\"0001-01-01T00:00:00.000Z\",\"transactionType\":\"ACCOUNT_DEBIT\",\"transactionId\":0,\"winLose\":
\"RESULT_ERR\"}"
                },
                "legacyData": {
                    "avgPrice": 0,
                    "betSize": 0,
                    "betType": "B",
                    "betCategoryType": "NONE",
                    "commissionRate": "5%",
                    "eventId": 102828127,
                    "eventTypeId": 1,
                    "fullMarketName": "Fixtures 03 June      / Landskrona v GAIS/ Match Odds",
                    "grossBetAmount": 18,
                    "marketName": "Match Odds",
                    "marketType": "O",
                    "placedDate": "2018-06-07T09:27:58.000Z",
                    "selectionId": 0,
                    "startDate": "0001-01-01T00:00:00.000Z",
                    "transactionType": "ACCOUNT_DEBIT",
                    "transactionId": 0,
                    "winLose": "RESULT_ERR"
                },
                "itemClass": "UNKNOWN"
            },
            {
                "refId": "10841763486",
                "itemDate": "2018-06-07T09:30:27.000Z",
                "amount": 18,
                "balance": 1000035.1,
                "itemClassData": {
                    "unknownStatementItem": "{\"avgPrice\":10.0,\"betSize\":2.0,\"betType\":\"B\",\"
betCategoryType\":\"E\",\"commissionRate\":null,\"eventId\":102828127,\"eventTypeId\":1,\"fullMarketName\":\"
Fixtures 03 June      / Landskrona v GAIS/ Match Odds\",\"grossBetAmount\":0.0,\"marketName\":\"Match Odds\",\"
marketType\":\"O\",\"placedDate\":\"2018-06-07T09:27:58.000Z\",\"selectionId\":130432,\"selectionName\":\"
Landskrona\",\"startDate\":\"0001-01-01T00:00:00.000Z\",\"transactionType\":\"ACCOUNT_CREDIT\",\"
transactionId\":0,\"winLose\":\"RESULT_ERR\"}"
                },
                "legacyData": {
                    "avgPrice": 10,
                    "betSize": 2,
                    "betType": "B",
                    "betCategoryType": "E",
                    "eventId": 102828127,
                    "eventTypeId": 1,
                    "fullMarketName": "Fixtures 03 June      / Landskrona v GAIS/ Match Odds",
                    "grossBetAmount": 0,
                    "marketName": "Match Odds",
                    "marketType": "O",
                    "placedDate": "2018-06-07T09:27:58.000Z",
                    "selectionId": 130432,
                    "selectionName": "Landskrona",
                    "startDate": "0001-01-01T00:00:00.000Z",
                    "transactionType": "ACCOUNT_CREDIT",
                    "transactionId": 0,
                    "winLose": "RESULT_ERR"
                },
                "itemClass": "UNKNOWN"
            },
            {
                "refId": "10841762387",
                "itemDate": "2018-06-07T09:14:03.000Z",
                "amount": -0.9,
                "balance": 1000051.3,
                "itemClassData": {
                    "unknownStatementItem": "{\"avgPrice\":0.0,\"betSize\":0.0,\"betType\":\"B\",\"
betCategoryType\":\"NONE\",\"commissionRate\":\"5%\",\"eventId\":102833966,\"eventTypeId\":7,\"fullMarketName\":
\"GB / Uttox  6th Jun/ 13:40 2m Hcap Hrd\",\"grossBetAmount\":18.0,\"marketName\":\"2m Hcap Hrd\",\"
marketType\":\"O\",\"placedDate\":\"2018-06-06T16:37:13.000Z\",\"selectionId\":0,\"selectionName\":null,\"
startDate\":\"2018-06-06T12:40:00.000Z\",\"transactionType\":\"ACCOUNT_DEBIT\",\"transactionId\":0,\"winLose\":
\"RESULT_NOT_APPLICABLE\"}"
                },
                "legacyData": {
                    "avgPrice": 0,
                    "betSize": 0,
                    "betType": "B",
                    "betCategoryType": "NONE",
                    "commissionRate": "5%",
                    "eventId": 102833966,
                    "eventTypeId": 7,
                    "fullMarketName": "GB / Uttox  6th Jun/ 13:40 2m Hcap Hrd",
                    "grossBetAmount": 18,
                    "marketName": "2m Hcap Hrd",
                    "marketType": "O",
                    "placedDate": "2018-06-06T16:37:13.000Z",
                    "selectionId": 0,
                    "startDate": "2018-06-06T12:40:00.000Z",
                    "transactionType": "ACCOUNT_DEBIT",
                    "transactionId": 0,
                    "winLose": "RESULT_NOT_APPLICABLE"
                },
                "itemClass": "UNKNOWN"
            },
            {
                "refId": "10841762387",
                "itemDate": "2018-06-07T09:14:03.000Z",
                "amount": 18,
                "balance": 1000052.2,
                "itemClassData": {
                    "unknownStatementItem": "{\"avgPrice\":10.0,\"betSize\":2.0,\"betType\":\"B\",\"
betCategoryType\":\"E\",\"commissionRate\":null,\"eventId\":102833966,\"eventTypeId\":7,\"fullMarketName\":\"GB
/ Uttox  6th Jun/ 13:40 2m Hcap Hrd\",\"grossBetAmount\":0.0,\"marketName\":\"2m Hcap Hrd\",\"marketType\":\"
O\",\"placedDate\":\"2018-06-06T16:37:13.000Z\",\"selectionId\":4700101,\"selectionName\":\"Catchin Time\",\"
startDate\":\"2018-06-06T12:40:00.000Z\",\"transactionType\":\"ACCOUNT_CREDIT\",\"transactionId\":0,\"winLose\":
\"RESULT_WON\"}"
                },
                "legacyData": {
                    "avgPrice": 10,
                    "betSize": 2,
                    "betType": "B",
                    "betCategoryType": "E",
                    "eventId": 102833966,
                    "eventTypeId": 7,
                    "fullMarketName": "GB / Uttox  6th Jun/ 13:40 2m Hcap Hrd",
                    "grossBetAmount": 0,
                    "marketName": "2m Hcap Hrd",
                    "marketType": "O",
                    "placedDate": "2018-06-06T16:37:13.000Z",
                    "selectionId": 4700101,
                    "selectionName": "Catchin Time",
                    "startDate": "2018-06-06T12:40:00.000Z",
                    "transactionType": "ACCOUNT_CREDIT",
                    "transactionId": 0,
                    "winLose": "RESULT_WON"
                },
                "itemClass": "UNKNOWN"
            }
        ],
        "moreAvailable": false
    }
}
```

Regulator Codes

The below regulator codes are used in the reglulators field provided in the listMarketCatalogue Market Description and the Exchange Stream API marketDefinition.

Abbreviation	Full Description
MR_ESP	SPANISH GAMBLING AUTHORITY
MR_INT	INDIAN MARKET REGULATOR
MR_ITA	AMMINISTRAZIONE AUTONOMA DEI MONOPOLI DI STATO
MR_NJ	NJRC - NEW JERSEY RACING COMMISSION

MR_TGC	THE TASMANIAN GAMING COMMISSION

EventTypeIds List

The below file contains a complete list of EventTypeIds

EventTypIds.xlsx

- Understanding Market Navigation
- Betfair Price Increments
- Common Error Codes
- Currency Parameters
- Locale Specification
- Racecourse Abbreviations
- Runner Metadata Description
- Time Zones & Time Format
- Virtual Bets
- Account Statement - Resettlement Lifecycle
- Regulator Codes
- EventTypeIds List

Reference Guide

Betting API Operations

Betting API Overview

Accounts API Operations

Accounts API Overview

Vendor API Operations

Vendor Services API

- getVendorClientId
- activateApplicationSubscription
- cancelApplicationSubscription
- updateApplicationSubscription
- getApplicationSubscriptionToken
- getApplicationSubscriptionHistory
- listAccountSubscriptionTokens
- listApplicationSubscriptionTokens
- isAccountSubscribedToWebApp
- token
- getVendorDetails
- revokeAccessToWebApp
- getAffiliateRelation

Betting API

- Endpoints
- Operation summary

Endpoints

Please find the details for the current Betting API endpoints.

If you have an **Italian** or **Spanish Exchange** account please see:

- **Betting on Italian Exchange**
- **Betting on Spanish Exchange**

Global Exchange

Interface	Endpoint	JSON-RPC Prefix	<method> Example
JSON-RPC	https://api.betfair.com/exchange/betting/json-rpc/v1	<method>	SportsAPING/v1.0/listMarketBook
JSON REST	https://api.betfair.com/exchange/betting/rest/v1.0/		listMarketBook/

Operation summary

Type	Operation	Description
List< EventTypeResult >	**listEventTypes (MarketFilter filter** ,Stringlocale **)**	Returns a list of Event Types (i.e. Sports) associated with the markets selected by the MarketFilter.
List< CompetitionResult >	**listCompetitions (MarketFilter filter** ,Stringlocale **)**	Returns a list of Competitions (i.e., World Cup 2013) associated with the markets selected by the MarketFilter. **Please note:** Specific eventTypes (horse racing & greyhounds) are not associated with specific competitions. These eventTypes are only associated with Venues (see listVenues)
List< TimeRangeResult >	**listTimeRanges (MarketFilter filter** , **TimeGranularity granularity)**	Returns a list of time ranges in the granularity specified in the request (i.e. 3PM to 4PM, Aug 14th to Aug 15th) associated with the markets selected by the MarketFilter.
List< EventResult >	**listEvents (MarketFilter filter** ,Stringlocale **)**	Returns a list of Events (i.e, Reading vs. Man United) associated with the markets selected by the MarketFilter.
List< MarketTypeResult >	**listMarketTypes (MarketFilter filter** ,Stringlocale **)**	Returns a list of market types (i.e. MATCH_ODDS, NEXT_GOAL) associated with the markets selected by the MarketFilter. The market types are always the same, regardless of locale.
List< CountryCodeResult >	**listCountries (MarketFilter filter** ,Stringlocale **)**	Returns a list of Countries associated with the markets selected by the MarketFilter.
List< VenueResult >	**listVenues (MarketFilter filter** ,Stringlocale **)**	Returns a list of Venues (i.e. Cheltenham, Ascot) associated with the markets selected by the MarketFilter. **Please note**: Only horse racing & greyhound markets are associated with a Venue.
List< MarketCatalogue >	**listMarketCatalogue (MarketFilter filter** ,Set< MarketProjection >marketProjection, MarketSort sort, int maxResults, Stringlocale **)**	Returns a list of information about published (ACTIVE/SUSPENDED) markets that does not change (or changes very rarely).
List< MarketBook >	**listMarketBook (List<String>marketIds** , PriceProjection priceProjection, OrderProjection orderProjection, MatchProjection matchProjection,StringcurrencyCode, Stringlocale, Date matchedSince, Set<BetId> betIds **)**	Returns a list of dynamic data about markets. Dynamic data includes prices, the status of the market, the status of selections, the traded volume, and the status of any orders you have placed in the market
	List<MarketBook> listRunnerBook (MarketId marketId, **SelectionId selectionId** , double handicap, PriceProjection priceProjection, OrderProjection orderProjection	Returns a list of dynamic data about **a market** and a **specified runner**. Dynamic data includes prices, the status

List<MarketBook>	, MatchProjection matchProjection, boolean includeOverallPosition, boolean partitionMatchedByStrategyRef, Set<String> customerStrategyRefs, StringcurrencyCode,Stringlocale, Date matchedSince, Set<BetId> betIds**) throws APINGException**	of the market, the status of selections, the traded volume, and the status of any orders you have placed in the market..
List<MarketProfitAndLoss>	**listMarketProfitAndLoss (Set<MarketId> marketIds**, boolean includeSettledBets, boolean includeBspBets, boolean netOfCommission **)**	Retrieve profit and loss for a given list of OPEN markets. The values are calculated using matched bets and optionally settled bets
CurrentOrderSummaryReport	**listCurrentOrders (Set<String>betIds**,Set<String>marketIds, OrderProjection orderProjection, TimeRangeplacedDateRange, OrderBy orderBy, SortDir sortDir, intfromRecord,intrecordCount **)**	Returns a list of your current orders.
ClearedOrderSummaryReport	**listClearedOrders (BetStatus betStatus**, Set<EventTypeId> eventTypeIds, Set<EventId> eventIds, Set<MarketId> marketIds, Set<RunnerId> runnerIds, Set< BetId> betIds, Side side, TimeRange settledDateRange, GroupBy groupBy, boolean includeItemDescription, String locale, int fromRecord, int recordCount **)**	Returns a list of settled bets based on the bet status, ordered by settled date
PlaceExecutionReport	**placeOrders (StringmarketId**, **List<PlaceInstruction> instructions**, StringcustomerRef, MarketVersion marketVersion, String customerStrategyRef, boolean async**)**	Place new orders into market.
CancelExecutionReport	**cancelOrders (StringmarketId**, **List<CancelInstruction>instructions**, StringcustomerRef **)**	Cancel all bets OR cancel all bets on a market OR fully or partially cancel particular orders on a market
ReplaceExecutionReport	**replaceOrders (StringmarketId**, **List< ReplaceInstruction> instructions**, StringcustomerRef, MarketVersion marketVersion, boolean async)	This operation is logically a bulk cancel followed by a bulk place. The cancel is completed first then the new orders are placed.
UpdateExecutionReport	**updateOrders (StringmarketId**, **List< UpdateInstruction> instructions**, StringcustomerRef **)**	Update non-exposure changing fields

- Betting on Spanish Exchange
- Betting On Italian Exchange

- Endpoints
- Operation summary

listEventTypes

Operation

listEventTypes

List< EventTypeResult > listEventTypes (MarketFilter filter ,Stringlocale **) throws APINGException**

Returns a list of Event Types (i.e. Sports) associated with the markets selected by the MarketFilter.

Parameter name	Type	Required	Description
filter	MarketFilter	✓	The filter to select desired markets. All markets that match the criteria in the filter are selected.
locale	String		The language used for the response. If not specified, the default is returned.

Return type	Description
List< EventTypeResult >	output data

Throws	Description
APINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

listCompetitions

Operation

listCompetitions

List< CompetitionResult > listCompetitions (MarketFilter filter ,Stringlocale **) throws APINGException**

Returns a list of Competitions (i.e., World Cup 2013) associated with the markets selected by the MarketFilter. Currently only Football markets have an associated competition.

Parameter name	Type	Required	Description
filter	MarketFilter	✓	The filter to select desired markets. All markets that match the criteria in the filter are selected.
locale	String		The language used for the response. If not specified, the default is returned.

Return type	Description
List< CompetitionResult >	output data

Throws	Description
APINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

listTimeRanges

Operation

listTimeRanges

List< TimeRangeResult > listTimeRanges (MarketFilter filter , TimeGranularity granularity) throws APINGException

Returns a list of time ranges in the granularity specified in the request (i.e. 3PM to 4PM, Aug 14th to Aug 15th) associated with the markets selected by the MarketFilter.

Parameter name	Type	Required	Description
filter	MarketFilter	✓	The filter to select desired markets. All markets that match the criteria in the filter are selected.
granularity	TimeGranularity	✓	The granularity of time periods that correspond to markets selected by the market filter.

Return type	Description
List< TimeRangeResult >	output data

Throws	Description
APINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

listEvents

Operation

listEvents

List< EventResult > listEvents (MarketFilter filter ,Stringlocale **) throws APINGException**

Returns a list of Events (i.e, Reading vs. Man United) associated with the markets selected by the MarketFilter.

Parameter name	Type	Required	Description
filter	MarketFilter	✓	The filter to select desired markets. All markets that match the criteria in the filter are selected.
locale	String		The language used for the response. If not specified, the default is returned.

Return type	Description
List< EventResult >	output data

Throws	Description
APINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

listMarketTypes

Operation

listMarketTypes

List< MarketTypeResult > listMarketTypes (MarketFilter filter ,Stringlocale **) throws APINGException**

Returns a list of market types (i.e. MATCH_ODDS, NEXT_GOAL) associated with the markets selected by the MarketFilter. The market types are always the same, regardless of locale.

Parameter name	Type	Required	Description
filter	MarketFilter	✓	The filter to select desired markets. All markets that match the criteria in the filter are selected.
locale	String		The language used for the response. If not specified, the default is returned.

Return type	Description
List< MarketTypeResult >	output data

Throws	Description
APINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

listCountries

Operation

listCountries

List< CountryCodeResult > listCountries (MarketFilter filter ,Stringlocale **) throws APINGException**

Returns a list of Countries associated with the markets selected by the MarketFilter.

Parameter name	Type	Required	Description
filter	MarketFilter	✓	The filter to select desired markets. All markets that match the criteria in the filter are selected.
locale	String		The language used for the response. If not specified, the default is returned.

Return type	Description
List< CountryCodeResult >	output data

Throws	Description
APINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

listVenues

Operation

listVenues

List< VenueResult > listVenues (MarketFilter filter ,Stringlocale **) throws APINGException**

Returns a list of Venues (i.e. Cheltenham, Ascot) associated with the markets selected by the MarketFilter. Currently, only Horse Racing markets are associated with a Venue.

Parameter name	Type	Required	Description
filter	MarketFilter	✓	The filter to select desired markets. All markets that match the criteria in the filter are selected.
locale	String		The language used for the response. If not specified, the default is returned.

Return type	Description
List< VenueResult >	output data

Throws	Description
APINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

listMarketCatalogue

Operation

listMarketCatalogue

List< MarketCatalogue > listMarketCatalogue (MarketFilter filter ,Set< MarketProjection >marketProjection, MarketSort sort, **intmaxResults** , Stringlocale **) throws APINGException**

Returns a list of information about published (ACTIVE/SUSPENDED) markets that does not change (or changes very rarely). You use listMarketCatalogue to retrieve the name of the market, the names of selections and other information about markets. Market Data Request Limits apply to requests made to listMarketCatalogue.

Please note: listMarketCatalogue does not return markets that are CLOSED.

Parameter name	Type	Required	Description
filter	MarketFilter	✓	The filter to select desired markets. All markets that match the criteria in the filter are selected.
marketProjection	Set< MarketProjection >		The type and amount of data returned about the market.
sort	MarketSort		The order of the results. Will default to **RANK** if not passed. **RANK** is an assigned priority that is determined by our Market Operations team in our back-end system. A result's overall rank is derived from the ranking given to the flowing attributes for the result. EventType, Competition, StartTime, MarketType, MarketId. For example, EventType is ranked by the most popular sports types and marketTypes are ranked in the following order: ODDS ASIAN LINE RANGE If all other dimensions of the result are equal, then the results are ranked in MarketId order.
maxResults	int	✓	limit on the total number of results returned, must be greater than 0 and less than or equal to 1000
locale	String		The language used for the response. If not specified, the default is returned.

Return type	Description
List< MarketCatalogue >	output data

Throws	Description
APINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

RUNNER_METADATA Description

The RUNNER_METADATA returned by listMarketCatalogue for **Horse Racing** (when available) is described in the table below.

Parameter	Description	Example
WEIGHT_UNITS	The unit of weight used	pounds
ADJUSTED_RATING	Adjusted ratings are race-specific ratings which reflect weights allocated in the race and, in some circumstances, the age of the horse. Collectively they represent the chance each runner has on form. https://www.timeform.com/Racing/Articles/How_the_ratings_for_a_race_are_calculated	79
DAM_YEAR_BORN	The year the horse's mother's birth	1997
DAYS_SINCE_LAST_RUN	The number of days since the horse last ran	66
WEARING	Any extra equipment the horse is wearing	tongue strap
DAMSIRE_YEAR_BORN	The year in which the horse's grandfather was born on its mothers side	1988
SIRE_BRED	The country were the horse's father was bred	IRE

TRAINER_NAME	The name of the horse's trainer	Fergal O'Brien
STALL_DRAW	The stall number the horse is starting from	10
SEX_TYPE	The sex of the horse	f
OWNER_NAME	The owner of the horse	Mr M. C. Fahy
SIRE_NAME	The name of the horse's father	Revoque
FORECAST PRICE_NUMERATOR	The forecast price numerator	13
FORECAST PRICE_DENOMINATOR	The forecast price denominator	8
JOCKEY_CLAIM	The reduction in the weight that the horse carries for a particular jockey	5
WEIGHT_VALUE	The weight of the horse	163
DAM_NAME	The name of the horse's mother	Rare Gesture
AGE	The age of the horse	7
COLOUR_TYPE	The colour of the horse	b
DAMSIRE_BRED	The country were the horse's grandfather was born	IRE
DAMSIRE_NAME	The name of the horse's grandfather	Shalford
SIRE_YEAR_BORN	The year the horse's father was born	1994
OFFICIAL_RATING	The horses official rating	97
FORM	The horses recent form	212246
BRED	The country in which the horse was born	IRE
runnerId	The runnerId for the horse	62434983
JOCKEY_NAME	The name of the jockey. **Please note**: This field will contain '**Reserve**' in the event that the horse has been entered into the market as a reserve runner. Any reserve runners will be withdrawn from the market once it has been confirmed that they will not run.	Paddy Brennan
DAM_BRED	The country where the horse's mother was born	IRE
COLOURS_DESCRIPTION	The textual description of the jockey silk	Royal blue and black check, white sleeves and cap
COLOURS_FILENAME	A relative URL to an image file corresponding to the jockey silk. You must add the value of this field to the base URL: http://content-cache.betfair.com/feeds_images/Horses/SilkColours/	c20140225lei/00058836.jpg
CLOTH_NUMBER	The number on the saddle-cloth	5
CLOTH_NUMBER ALPHA	The number on the saddle-cloth. For US Racing were the runner is paired, this field will display the cloth number of the paired runner e.g. "1A"	

listMarketBook

Operation

listMarketBook

List< MarketBook > listMarketBook (List<String>marketIds , PriceProjection priceProjection, OrderProjection orderProjection, MatchProjection matchProjection, boolean includeOverallPosition, boolean partitionMatchedByStrategyRef, Set<String> customerStrategyRefs, StringcurrencyCode, Stringlocale, Date matchedSince, Set<BetId> betIds**) throws APINGException**

Returns a list of dynamic data about markets. Dynamic data includes prices, the status of the market, the status of selections, the traded volume, and the status of any orders you have placed in the market.

Please note: Separate requests should be made for **OPEN & CLOSED** markets. Request that include both **OPEN & CLOSED** markets will only return those markets that are **OPEN.**

Market Data Request Limits apply to requests made to **listMarketBook** that include price or order projections.

Calls to **listMarketBook** should be made up to a maximum of 5 times per second to a single marketId.
Best Practice

Customers seeking to use listMarketBook to obtain price, volume, unmatched **(EXECUTABLE)** orders and matched position in a single operation should provide an **OrderProjection**of **"EXECUTABLE"** in their listMarketBook request and receive all unmatched **(EXECUTABLE)** orders and the aggregated matched volume from all orders irrespective of whether they are partially or fully matched. The level of matched volume aggregation (**Match Projection**) requested should be **ROLLED_UP_BY_AVG_PRICE** or **ROLLED_UP_BY_PRICE**, the former being preferred. This provides a single call in which you can track prices, traded volume, unmatched orders and your evolving matched position with a reasonably fixed, minimally sized response.

Parameter name	Type	Required	Description
marketIds	List<String>	✓	One or more market ids. The number of markets returned depends on the amount of data you request via the price projection.
priceProjection	PriceProjection		The projection of price data you want to receive in the response.
orderProjection	OrderProjection		The orders you want to receive in the response.
matchProjection	MatchProjection		If you ask for orders, specifies the representation of matches.
includeOverallPosition	boolean		If you ask for orders, returns matches for each selection. Defaults to true if unspecified.
partitionMatchedByStrategyRef	boolean		If you ask for orders, returns the breakdown of matches by strategy for each selection. Defaults to false if unspecified.
customerStrategyRefs	Set<String>		If you ask for orders, restricts the results to orders matching any of the specified set of customer defined strategies. Also filters which matches by strategy for selections are returned, if partitionMatchedByStrategyRef is true. An empty set will be treated as if the parameter has been omitted (or null passed).
currencyCode	String		A Betfair standard currency code. If not specified, the default currency code is used.
locale	String		The language used for the response. If not specified, the default is returned.
matchedSince	Date		If you ask for orders, restricts the results to orders that have at least one fragment matched since the specified date (all matched fragments of such an order will be returned even if some were matched before the specified date). All EXECUTABLE orders will be returned regardless of matched date.
betIds	Set<BetId>		If you ask for orders, restricts the results to orders with the specified bet IDs. Omitting this parameter means that all bets will be included in the response. **Please note:** A maximum of 250 betId's can be provided at a time.

Return type	Description
List< MarketBook >	output data

Throws	Description
APINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

listRunnerBook

List<MarketBook> listRunnerBook (MarketId marketId, SelectionId selectionId, double handicap, PriceProjection priceProjection, OrderProjection orderProjection, MatchProjection matchProjection, boolean includeOverallPosition, boolean partitionMatchedByStrategyRef, Set<String> customerStrategyRefs, StringcurrencyCode,Stringlocale, Date matchedSince, Set<BetId> betIds**) throws APINGException**

Returns a list of dynamic data about **a market** and a **specified runner**. Dynamic data includes prices, the status of the market, the status of selections, the traded volume, and the status of any orders you have placed in the market..

listRunnerBook behaviour

You can only pass in one marketId and one selectionId in that market per request. If the selectionId being passed in is not a valid one / doesn’t belong in that market then the call will still work but only the market data is returned

Parameter name	Type	Required	Description
marketId	MarketId	✓	The unique id for the market..
selectionId	SelectionId	✓	The unique id for the selection in the market.
handicap	double		The projection of price data you want to receive in the response.
priceProjection	PriceProjection		The projection of price data you want to receive in the response.
orderProjection	OrderProjection		The orders you want to receive in the response.
matchProjection	MatchProjection		If you ask for orders, specifies the representation of matches.
includeOverallPosition	boolean		If you ask for orders, returns matches for each selection. Defaults to true if unspecified.
partitionMatchedByStrategyRef	boolean		If you ask for orders, returns the breakdown of matches by strategy for each selection. Defaults to false if unspecified.
customerStrategyRefs	Set<String>		If you ask for orders, restricts the results to orders matching any of the specified set of customer defined strategies. Also filters which matches by strategy for selections are returned, if partitionMatchedByStrategyRef is true. An empty set will be treated as if the parameter has been omitted (or null passed).
currencyCode	String		A Betfair standard currency code. If not specified, the default currency code is used.
locale	String		The language used for the response. If not specified, the default is returned.
matchedSince	Date		If you ask for orders, restricts the results to orders that have at least one fragment matched since the specified date (all matched fragments of such an order will be returned even if some were matched before the specified date). All EXECUTABLE orders will be returned regardless of matched date.
betIds	Set<BetId>		If you ask for orders, restricts the results to orders with the specified bet IDs. Omitting this parameter means that all bets will be included in the response. **Please note:** A maximum of 250 betId's can be provided at a time.

Return type	Description
List< MarketBook >	output data

Throws	Description
APINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

listMarketProfitAndLoss

listMarketProfitAndLoss

List<MarketProfitAndLoss> listMarketProfitAndLoss (Set<MarketId> marketIds, boolean includeSettledBets, boolean includeBspBets, boolean netOfCommission **) throws APINGException**

Retrieve profit and loss for a given list of OPEN markets. The values are calculated using matched bets and optionally settled bets. **Only odds (MarketBettingType = ODDS) markets are implemented, markets of other types are silently ignored.**

To retrieve your profit and loss for CLOSED markets, please use the listClearedOrders request.

Please note: Market Data Request Limits apply to requests made to listMarketProfitAndLoss

Parameter name	Type	Required	Description
marketIds	Set<MarketId>	✓	List of markets to calculate profit and loss
includeSettledBets	boolean		Option to include settled bets (partially settled markets only). Defaults to false if not specified.
includeBspBets	boolean		Option to include BSP bets. Defaults to false if not specified.
netOfCommission	boolean		Option to return profit and loss net of users current commission rate for this market including any special tariffs. Defaults to false if not specified.

Return type	Description
List<MarketProfitAndLoss>	

Throws	Description
APINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

listCurrentOrders

Operation

listCurrentOrders

CurrentOrderSummaryReport listCurrentOrders (Set<String>betIds,Set<String>marketIds, OrderProjection orderProjection, TimeRange placedDateRange, TimeRange dateRange, OrderBy orderBy, SortDir sortDir,intfromRecord,intrecordCount **) throws APINGException**

Returns a list of your current orders. Optionally you can filter and sort your current orders using the various parameters, setting none of the parameters will return all of your current orders up to a maximum of 1000 bets, ordered BY_BET and sorted EARLIEST_TO_LATEST. To retrieve more than 1000 orders, you need to make use of the fromRecord and recordCount parameters.

Best Practice

To efficiently track new bet matches from a specific time, customers should use a combination of the **dateRange**, **orderBy "BY_MATCH_TIME"** and **orderProjection “ALL”** to filter fully/partially matched orders from the list of returned bets. The response will then filter out any bet records that have no matched date and provide a list of betIds in the order which they are fully/partially matched from the date and time specified in the dateRange field.

Parameter name	Type	Required	Description
betIds	Set<String>		Optionally restricts the results to the specified bet IDs. A maximum of 250 betId's, or a combination of 250 betId's & marketId's are permitted.
marketIds	Set<String>		Optionally restricts the results to the specified market IDs. A maximum of 250 marketId's, or a combination of 250 marketId's & betId's are permitted.
orderProjection	OrderProjection		Optionally restricts the results to the specified order status.
customerOrderRefs	Set<CustomerOrderRef>		Optionally restricts the results to the specified customer order references.
customerStrategyRefs	Set<CustomerStrategyRef>		Optionally restricts the results to the specified customer strategy references.
dateRange	TimeRange		Optionally restricts the results to be from/to the specified date, these dates are contextual to the orders being returned and therefore the dates used to filter on will change to placed, matched, voided or settled dates depending on the orderBy. This date is inclusive, i.e. if an order was placed on exactly this date (to the millisecond) then it will be included in the results. If the from is later than the to, no results will be returned.
orderBy	OrderBy		Specifies how the results will be ordered. If no value is passed in, it defaults to BY_BET. Also acts as a filter such that only orders with a valid value in the field being ordered by will be returned (i.e. BY_VOID_TIME returns only voided orders, BY_SETTLED_TIME (applies to partially settled markets) returns only settled orders and BY_MATCH_TIME returns only orders with a matched date (voided, settled, matched orders)). Note that specifying an orderBy parameter defines the context of the date filter applied by the dateRange parameter (placed, matched, voided or settled date) - see the dateRange parameter description (above) for more information. See also the OrderBy type definition.
sortDir	SortDir		Specifies the direction the results will be sorted in. If no value is passed in, it defaults to EARLIEST_TO_LATEST.
fromRecord	int		Specifies the first record that will be returned. Records start at index zero, not at index one.
recordCount	int		Specifies how many records will be returned from the index position 'fromRecord'. Note that there is a page size limit of 1000. A value of zero indicates that you would like all records (including and from 'fromRecord') up to the limit.

Return type	Description
CurrentOrderSummaryReport	

Throws	Description
APINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

listClearedOrders

Operation

listClearedOrders

ClearedOrderSummaryReport listClearedOrders (BetStatus betStatus, Set<EventTypeId> eventTypeIds, Set<EventId> eventIds, Set<MarketId> marketIds, Set<RunnerId> runnerIds, Set<BetId> betIds, Side side, TimeRange settledDateRange, GroupBy groupBy, boolean includeItemDescription, String locale, int fromRecord, int recordCount **) throws APINGException**

Returns a list of settled bets based on the bet status, ordered by settled date. To retrieve more than 1000 records, you need to make use of the fromRecord and recordCount parameters. By default the service will return all available data for the last 90 days (see Best Practice note below). The fields available at each roll-up are available here

Best Practice

ⓘ You should specify a **settledDateRange "from"** date when making requests for data. This reduces the amount of data that requires downloading & improves the speed of the response. Specifying a "from" date of the last call will ensure that only new data is returned.

Parameter name	Type	Required	Description
betStatus	BetStatus	✓	Restricts the results to the specified status.
eventTypeIds	Set<EventTypeId>		Optionally restricts the results to the specified Event Type IDs.
eventIds	Set<EventId>		Optionally restricts the results to the specified Event IDs.
marketIds	Set<Market Id>		Optionally restricts the results to the specified market IDs.
runnerIds	Set<RunnerId>		Optionally restricts the results to the specified Runners.
betIds	Set<BetId>		Optionally restricts the results to the specified bet IDs. A maximum of 1000 betId's are allowed in a single request.
customerOrderRefs	Set<CustomerOrderRef>		Optionally restricts the results to the specified customer order references.
customerStrategyRefs	Set<CustomerStrategyRef>		Optionally restricts the results to the specified customer strategy references.
side	Side		Optionally restricts the results to the specified side.
settledDateRange	TimeRange		Optionally restricts the results to be from/to the specified settled date. This date is inclusive, i.e. if an order was cleared on exactly this date (to the millisecond) then it will be included in the results. If the from is later than the to, no results will be returned. **Please note:** This is ignored when using groupBy MARKET
groupBy	GroupBy		How to aggregate the lines, if not supplied then the lowest level is returned, i.e. bet by bet This is only applicable to SETTLED BetStatus.
includeItemDescription	boolean		If true then an ItemDescription object is included in the response.
locale	String		The language used for the itemDescription. If not specified, the customer account default is returned.
fromRecord	int		Specifies the first record that will be returned. Records start at index zero.
recordCount	int		Specifies how many records will be returned, from the index position 'fromRecord'. Note that there is a page size limit of 1000. A value of zero indicates that you would like all records (including and from 'fromRecord') up to the limit.

Return type	Description
ClearedOrderSummaryReport	

Throws	Description
APINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

listClearedOrders - Roll-up Fields Available

The below table indicates fields will be available at each roll-up when making requests to **listClearedOrders** using the **groupBy** parameter.

listClearedOrders will 'hoist' data into higher rollup levels whenever the value is unambiguous, this can occur when,

- SIDE, RUNNER & MARKET level when only 1 bet is involved in the rollup - only 1 possible value for each field so they are all unambiguous (in particular, betId)
- RUNNER level when all bets are on the same side - value of side is unambiguous, priceRequested and priceMatched can be averaged, sizeSettled can be totalled.
- MARKET level when all bets are on the same selection/side combo - value of selectionId, handicap and side are unambiguous, priceRequested and priceMatched can be averaged, sizeSettled can be totalled.
- PersistenceType and OrderType are only displayed above BET level if all bets in the rollup have the same type

*=fields that may be hoisted (if lower level fields are unambiguous or the rollup contains only 1 bet)

Rollup level:	**BET**	**SIDE**	**MARKET**	**EVENT**	**EVENT_TYPE**	**EXCHANGE**
Settled As	Y	Y	Y	Y	Y	Y
Settled Date	Y	MAX	MAX	MAX	MAX	MAX
Bet Count	Y	Y	Y	Y	Y	Y
Profit	Y	SUM	SUM	SUM	SUM	SUM
Exchange Id	Y	Y	Y	Y	Y	Y
Event Type Id	Y	Y	Y	Y	Y	N
Event Id	Y	Y	Y	Y	N	N
Market Id	Y	Y	Y	N	N	N
Selection Id	Y	Y	N*	N	N	N
Handicap	Y	Y	N*	N	N	N
Side	Y	Y	N*	N	N	N
Price Requested	Y	AVG	N*$^{\text{(AVG)}}$	N	N	N
Price Matched	Y	AVG	N*$^{\text{(AVG)}}$	N	N	N
Size Settled	Y	SUM	N*$^{\text{(SUM)}}$	N	N	N
Price Reduced	Y	Y	Y	N	N	N
Commission	N	N	Y	SUM	SUM	SUM
Bet Id	Y	N*	N*	N	N	N
Placed Date	Y	MAX	MAX	N	N	N
Persistence Type	Y	Y	Y	N	N	N
Order Type	Y	Y	Y	N	N	N
Regulator Code	Y	Y	Y	N	N	N
Regulator Auth Code	Y	Y	Y	N	N	N
Voided Date(where applicable)	Y	MAX	MAX	N	N	N
BetOutcome	Y	N	N	N	N	N

placeOrders

Operation

placeOrders

PlaceExecutionReport placeOrders (StringmarketId , List< PlaceInstruction >instructions, String customerRef, MarketVersion marketVersion, String customerStrategyRef, boolean async) throws APINGException

Place new orders into market. Please note that additional bet sizing rules apply to bets placed into the Italian Exchange.

In normal circumstances the placeOrders is an atomic operation. PLEASE NOTE: if the 'Best Execution' features is switched off, placeOrders can return 'PROCESSED_WITH_ERRORS' meaning that some bets can be rejected and other placed when submitted in the same PlaceInstruction

Parameter name	Type	Required	Description
marketId	String	✓	The market id these orders are to be placed on
instructions	List< PlaceInstruction >	✓	The number of place instructions. The limit of place instructions per request is 200 for the Global Exchange and 50 for the Italian Exchange.
customerRef	String		Optional parameter allowing the client to pass a unique string (up to 32 chars) that is used to de-dupe mistaken re-submissions. CustomerRef can contain: upper/lower chars, digits, chars : - . _ + * : ; ~ only. **Please note:** There is a time window associated with the de-duplication of duplicate submissions which is 60 seconds.
marketVersion	MarketVersion		Optional parameter allowing the client to specify which version of the market the orders should be placed on. If the current market version is higher than that sent on an order, the bet will be lapsed.
customerStrategyRef	String		An optional reference customers can use to specify which strategy has sent the order. The reference will be returned on order change messages through the stream API. The string is limited to 15 characters. If an empty string is provided it will be treated as null.
async	boolean		An optional flag (not setting equates to false) which specifies if the orders should be placed asynchronously. Orders can be tracked via the Exchange Stream API or or the API-NG by providing a customerOrderRef for each place order. An order's status will be PENDING and no bet ID will be returned. This functionality is available for all bet types - including Market on Close and Limit on Close

Return type	Description
PlaceExecutionReport	

Throws	Description
APINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

Placing a Bet

To place a bet you require the marketId and selectionId parameters from the listMarketCatalogue API call. The below parameters will place a normal Exchange bet at odds of 3.0 for a stake of £2.0.

If the bet is placed successfully, a betId is returned in the placeOrders response

placeOrders Request

```
[
    {
        "jsonrpc": "2.0",
        "method": "SportsAPING/v1.0/placeOrders",
        "params": {
            "marketId": "1.109850906",
            "instructions": [
                {
                    "selectionId": "237486",
                    "handicap": "0",
                    "side": "LAY",
                    "orderType": "LIMIT",
                    "limitOrder": {
                        "size": "2",
                        "price": "3",
                        "persistenceType": "LAPSE"
                    }
                }
            ]
        },
        "id": 1
    }
]
```

placeOrder Response

```
[
    {
        "jsonrpc": "2.0",
        "result": {
            "marketId": "1.109850906",
            "instructionReports": [
                {
                    "instruction": {
                        "selectionId": 237486,
                        "handicap": 0,
                        "limitOrder": {
                            "size": 2,
                            "price": 3,
                            "persistenceType": "LAPSE"
                        },
                        "orderType": "LIMIT",
                        "side": "LAY"
                    },
                    "betId": "31242604945",
                    "placedDate": "2013-10-30T14:22:47.000Z",
                    "averagePriceMatched": 0,
                    "sizeMatched": 0,
                    "status": "SUCCESS"
                }
            ],
            "status": "SUCCESS"
        },
        "id": 1
    }
]
```

Placing a Betfair SP Bet

To place a bet on a selection at Betfair SP, you need to specify the parameters below in the placeOrders request. The below example would place a Betfair SP back bet on the required selection for a stake of £2.00.

placeOrders Request

```
[
    {
        "jsonrpc": "2.0",
        "method": "SportsAPING/v1.0/placeOrders",
        "params": {
            "marketId": "1.111836557",
            "instructions": [
                {
                    "selectionId": "5404312",
                    "handicap": "0",
                    "side": "BACK",
                    "orderType": "MARKET_ON_CLOSE",
                    "marketOnCloseOrder": {
                        "liability": "2"
                    }
                }
            ]
        },
        "id": 1
    }
]
```

placeOrders Response

```
[
    {
        "jsonrpc": "2.0",
        "result": {
            "marketId": "1.111836557",
            "instructionReports": [
                {
                    "instruction": {
                        "selectionId": 5404312,
                        "handicap": 0,
                        "marketOnCloseOrder": {
                            "liability": 2
                        },
                        "orderType": "MARKET_ON_CLOSE",
                        "side": "BACK"
                    },
                    "betId": "31645233727",
                    "placedDate": "2013-11-12T12:07:29.000Z",
                    "status": "SUCCESS"
                }
            ],
            "status": "SUCCESS"
        },
        "id": 1
    }
]
```

Placing a 'Keep' Bet

To place a bet that will be kept once a market turns in-play (if unmatched), you must include the below parameters i.e. "persistenceType": "PERSIST"

The bet will then be placed automatically into the in-play market at the start of the event.

```
[
    {
        "jsonrpc": "2.0",
        "method": "SportsAPING/v1.0/placeOrders",
        "params": {
            "marketId": "1.109850906",
            "instructions": [
                {
                    "selectionId": "237486",
                    "handicap": "0",
                    "side": "LAY",
                    "orderType": "LIMIT",
                    "limitOrder": {
                        "size": "2",
                        "price": "3",
                        "persistenceType": "PERSIST"
                    }
                }
            ]
        },
        "id": 1
    }
]
```

```
[
    {
        "jsonrpc": "2.0",
        "result": {
            "marketId": "1.109850906",
            "instructionReports": [
                {
                    "instruction": {
                        "selectionId": 237486,
                        "handicap": 0,
                        "limitOrder": {
                            "size": 2,
                            "price": 3,
                            "persistenceType": "PERSIST"
                        },
                        "orderType": "LIMIT",
                        "side": "LAY"
                    },
                    "betId": "31242604945",
                    "placedDate": "2013-10-30T14:22:47.000Z",
                    "averagePriceMatched": 0,
                    "sizeMatched": 0,
                    "status": "SUCCESS"
                }
            ],
            "status": "SUCCESS"
        },
        "id": 1
    }
]
```

Betting Enhancements

Fill or Kill bets

By setting the optional parameter ‘**TimeInForce**’ on a **limitOrder** submission to the value ‘FILL_OR_KILL’ and optionally passing a **minFillSize** value, the Exchange will only match the order if at least the specified **minFillSize** can be matched (if passed) or the whole order matched (if not). Any order which cannot be so matched, and any remaining unmatched part of the order (if minFillSize is specified) will be immediately cancelled.

Please note: the matching algorithm for Fill or Kill orders behaves slightly differently to that for standard limit orders. Whereas the price on a limit order represents the lowest price at which any fragment should be matched, the price on a Fill or Kill order represents the lower limit of the Volume Weighted Average Price (“VWAP”) for the entire volume matched. So, for instance, a Fill or Kill order with price = 5.4 and size = 10 might be matched as £2 @ 5.5, £6 @ 5.4 and £2 @ 5.3.

Examples

FILL_OR_KILL order which was lapsed:

Request
[{"jsonrpc": "2.0", "method": "SportsAPING/v1.0/placeOrders", "params": {"marketId":"1.126124422","instructions":[{"selectionId":"10590221","handicap":"0","side":"BACK","orderType":"LIMIT","limitOrder":{"size":"5","price":"21","persistenceType":"LAPSE","timeInForce":"FILL_OR_KILL","minFillSize":"5"}}]}, "id": 1}]

Response
[{"jsonrpc":"2.0","result":{"status":"SUCCESS","marketId":"1.126124422","instructionReports":[{"status":"SUCCESS","instruction":{"selectionId":10590221,"handicap":0.0,"limitOrder":{"size":5.0,"price":21.0,"minFillSize":5.0,"timeInForce":"FILL_OR_KILL"},"orderType":"LIMIT","side":"BACK"},"betId":"72666364933","placedDate":"2016-08-12T10:45:15.000Z","averagePriceMatched":0.0,"sizeMatched":0.0}]},"id":1}]

FILL_AND_KILL request a minimum fill size of 3.00:

Request
[{"jsonrpc": "2.0", "method": "SportsAPING/v1.0/placeOrders", "params": {"marketId":"1.126124422","instructions":[{"selectionId":"10590221","handicap":"0","side":"BACK","orderType":"LIMIT","limitOrder":{"size":"5","price":"21","persistenceType":"LAPSE","timeInForce":"FILL_OR_KILL","minFillSize":"3"}}]}, "id": 1}]

Response
[{"jsonrpc":"2.0","result":{"status":"SUCCESS","marketId":"1.126124422","instructionReports":[{"status":"SUCCESS","instruction":{"selectionId":10590221,"handicap":0.0,"limitOrder":{"size":5.0,"price":21.0,"minFillSize":3.0,"timeInForce":"FILL_OR_KILL"},"orderType":"LIMIT","side":"BACK"},"betId":"72666433304","placedDate":"2016-08-12T10:47:42.000Z","averagePriceMatched":21.32267441860465,"sizeMatched":3.4400000000000004}]},"id":1}]

Market version parameter

We have added an additional optional parameter ‘**marketVersion**’ to the ‘**placeOrders**’ and ‘**replaceOrders**’ operations. The MarketBook data item, which contains the dynamic data on a market, including its prices, has always returned an integer market ‘version’. This ‘version’ is incremented when significant events – runner removal, turn in-play etc. – occur. Now, by passing that version as ‘marketVersion’ with your orders, you can specify that if the market version has been incremented beyond that value, your orders should lapse and not be submitted for matching.

This functionality should be of use to those who want to bet right up to the actual ‘off’ of a horse race or sporting event but be confident that you’re not inadvertently bet into the first seconds of in-play after the off. Similarly, in managed football markets, you can avoid your bets reaching the Exchange after the market has reformed following a goal being scored etc.

Notes on 'version' behavior

The ‘market version’ value (on listMarketBook and on ESA) is incremented for any and all changes to the market.

However, to prevent falsely blocking bets we keep track of the last material change (which we define as one performed under suspension**) and will only accept bets placed with that version or later.

	Market Version	Minimum version to not be rejected	Expected behaviour

Market Activated	1234	1234	
Start time updated (market not suspended)	1235	1234	Non-material change, bets placed before change but received after will be processed normally
Runner removed (under suspension)	1236	1236	Material change, bets placed before change but received after will be rejected.
Market turned in-play	1237	1237	As above

** this includes:

- **Runner removal and addition**
- **Turn in-play**
- **Lapsing or voiding bets (eg on goals being scored in managed Football** market)

But not (e.g.) updating Tennis court times or Golf tee times as they become more accurately known on the day.

Example of a request including market version:

```
[{"jsonrpc": "2.0", "method": "SportsAPING/v1.0/placeOrders", "params": {"marketId":"1.126086207","instructions":[{"selectionId":"63908","handicap":"0","
side":"BACK","orderType":"LIMIT","limitOrder":{"size":"2","price":"30"}}],"marketVersion":{"version":"123456789"}}, "id": 1}]
```

Bet to Payout or Profit/Liability

Place a bet specifying your target payout, profit or liability, instead of the backers stake ('size').

Currently, best execution, which guarantees that you'll receive the best possible price, means that you receive a greater potential payout to the same stakes (or risk a smaller potential payout to the same backer's stakes, for layers).

If you wish to benefit by receiving the same potential payout as you originally requested, but to smaller stakes, you can now specify on a LimitOrder (place Orders) an optional 'betTargetType' of 'PAYOUT' or 'BACKERS_PROFIT' (the latter being identical to layers' liability) and a 'betTargetSize' representing the value of that payout or profit, together with the usual 'price' parameter to represent their limit price. Your bet will then be matched to achieve that payout or profit at the specified price or better.

Should all or any of the order be unmatched after first reaching the Exchange, the unmatched portion will be expressed in standard price and backers' stake terms (by dividing the remaining unmatched payout by the price, or unmatched profit by the price – 1, and placed on the unmatched queue), after this point the bet behaves like any other.

Please note: This function is only enabled for UK & International customers and not .it, .es, .dk and .se jurisdictions.

Examples

Placing a back bet targeting a £2 profit

Request

```
[{"jsonrpc": "2.0", "method": "SportsAPING/v1.0/placeOrders", "params": {"marketId":"1.126124417","instructions":[{"selectionId":"11166583","
handicap":"0","side":"BACK","orderType":"LIMIT","limitOrder":{"price":"2","betTargetType":"BACKERS_PROFIT","betTargetSize":"2"}}]}, "id": 1}]
```

Response

```
 [{"jsonrpc":"2.0","result":{"status":"SUCCESS","marketId":"1.126124417","instructionReports":[{"status":"SUCCESS","instruction":{"selectionId":
11166583,"handicap":0.0,"limitOrder":{"price":2.0,"betTargetSize":2.0,"persistenceType":"LAPSE","betTargetType":"BACKERS_PROFIT"},"orderType":"
LIMIT","side":"BACK"},"betId":"72671225671","placedDate":"2016-08-12T13:00:23.000Z","averagePriceMatched":6.3999999999999995,"sizeMatched":
0.37}]},"id":1}]
```

Placing a lay bet targeting a £10 Payout

Request

```
[{"jsonrpc": "2.0", "method": "SportsAPING/v1.0/placeOrders", "params": {"marketId":"1.126122473","instructions":[{"selectionId":"11576316","
handicap":"0","side":"LAY","orderType":"LIMIT","limitOrder":{"price":"10","betTargetType":"PAYOUT","betTargetSize":"10"}}]}, "id": 1}]
```

Response

```
[{"jsonrpc":"2.0","result":{"status":"SUCCESS","marketId":"1.126122473","instructionReports":[{"status":"SUCCESS","instruction":{"selectionId":
11576316,"handicap":0.0,"limitOrder":{"price":10.0,"betTargetSize":10.0,"persistenceType":"LAPSE","betTargetType":"PAYOUT"},"orderType":"LIMIT","
side":"LAY"},"betId":"72671531256","placedDate":"2016-08-12T13:05:45.000Z","averagePriceMatched":4.2,"sizeMatched":2.38}]},"id":1}]
```

Ability to place lower minimum stakes at larger prices

In order to allow customers to bet to smaller stakes on longer-priced selections, an extra property has been added to our Currency Parameters – “Min Bet Payout”.

As currently LIMIT bets where the backer’s stake is at and above the ‘Min Bet Size’ for the currency concerned (£2 for GBP) are valid.   In addition, bets below this value are valid if the payout of the bet would be equal to or greater than the value of ‘**Min Bet Payout’** - £10 for GBP. For example, a bet of £1 @ 10, or 10p @ 100 or 1p @ 1000 are all valid as they all target a payout of £10 or more.

This functionality is available to "orderType: LIMIT" bets only.

Please note: This function is only enabled for UK & International customers and not .it, .es, .dk and .se jurisdictions.

Each Way Betting

Each Way betting is available via the API. Each Way markets can be identified as marketType EACH_WAY using listMarketCatalogue.

Please note: your potential liability for an each way bet is 'size' x 2, so for a 10 size bet your available to bet balance will be reduced by 20, for example.

The divisor that applies to the EACH_WAY market is returned by listMarketCatalogue via the MARKET_DESCRIPTION MarketProjection.

Please see table a table that indicates how the “Each-Way divisor" is determined for specific race types:

Race Type	Number of Runners(1)	Number of Places	Fraction of Win Odds “Each-Way divisor"
Handicap	16 or more	4	1/4
Handicap	12 to 15	3	1/4
Handicap	8 to 11	3	1/5
Handicap	5 to 7	2	1/4
Non-Handicap	8 or more	3	1/5
Non-Handicap	5 to 7	2	1/4

(1)   Number of Runners at Market creation time; place terms are fixed for the life of the market (like Betfair Place market) not dependent on number of runners at the off (like Fixed Odds EW markets)

We will not offer EW markets if the number of runners at market creation time is 4 or fewer

Betfair Price Increments

Below is a list of price increments per price 'group'. Placing a bet outside of these increments will result in an **INVALID_ODDS** error

Odds Markets

Price	Increment
1.01 2	0.01
2 3	0.02
3 4	0.05
4 6	0.1
6 10	0.2
10 20	0.5
20 30	1
30 50	2
50 100	5
100 1000	10

Asian Handicap & Total Goal Markets

Price	Increment
1.01 1000	0.01

Currency Parameters

Guide to **available currencies and minimum bet sizes**.

Betfair Starting Price Betting (BSP)

The Betfair Starting Price will be determined by balancing bets from customers who want to back and lay at Starting Price and matching into the Betfair exchange markets to balance out any residual demand.

The Betfair Starting Price will be calculated exactly to ensure the fairest and most transparent odds possible for both backers and layers. The BSP does not need to account for a profit margin but instead is calculated at the start of an event by looking at the relationship between the amounts of money requested at SP by opposing betting parties. To give an even more accurate price, we will use money where possible that is trading on the exchange at the start of the event. This gives a true reflection of public opinion on a selection.

How is the BSP calculated?

The Near Price is based on money currently on the site at SP as well as unmatched money on the same selection in the exchange. To understand this properly, you first need to understand the calculation of the Far Price, which only takes into account the SP bets that have been made. The Far Price is not as complicated but not as accurate and only accounts for money on the site at SP.

Excluding money requested at a fixed price on the exchange, if there are £1000 worth of backers stakes on a selection at SP and £6000 worth of layers liability, we can return an SP at the start of the event of 6/1 (7.0).

If however there were £6000 worth of backers stakes on the selection and £1000 worth of layers liability, we would return an SP of 1/6 (1.17). These are calculations of the Far Price.

The calculation of the final starting price occurs when the market is turned in-play. This is when the market is reconciled.

Further information and detailed working demonstrating how the Betfair SP price is calculated can be found via https://promo.betfair.com/betfairsp/FAQs_theBasics.html

Betting On Italian Exchange

How to Access the Italian Exchange API

Italian residents who have registered an Italian Exchange account can access the Betfair Exchange API. Italian residents can register an account via **https://register.betfair.it/account/registration**

To use the Italian Exchange API you need to **create an Application Key** for your Italian Exchange account.

Once you have done created an App Key, you will need to login to the Exchange API using the appropriate Italian Exchange endpoint which are as follows:

Non-interactive Login

For fully automated applications

```
https://identitysso-cert.betfair.it/api/certlogin
```

Interactive Login - Desktop Application

```
https://identitysso.betfair.it/view/login?product=<AppKey>&url=https://www.betfair.it
```

API Login - Desktop Application

```
https://identitysso.betfair.it/api/login
```

Once a session token has been obtained via the .it login methods above, any further API requests should be sent to the UK Exchange endpoints indicated below.

Requests to these endpoints will automatically return the markets that are available to Italian Exchange customers.

Betting API Endpoints

Interface	Endpoint	JSON-RPC Prefix	<methodname> Example
JSON-RPC	https://api.betfair.com/exchange/betting/json-rpc/v1	<methodname>	SportsAPING/v1.0/listMarketBook
JSON REST	https://api.betfair.com/exchange/betting/rest/v1.0/		

Accounts API Endpoints

Interface	Endpoint	JSON-RPC Prefix	<methodname> Example
JSON-RPC	https://api.betfair.com/exchange/account/json-rpc/v1	<methodname>	AccountAPING/v1.0/getAccountFunds
JSON REST	https://api.betfair.com/exchange/account/rest/v1.0		

Italian Exchange Specific Bet Rules

To use the interactive login with the Italian Exchange your App Key will need to be white-listed by Betfair. White-listing is not required to use the non-interactive login method.

There are several additional regulatory rules which apply specifically and only to accounts betting on the Italian Exchange (https://www.betfair.it/exchange):

1. The stake for each back offer is a minimum of 200 Euro Cents and can only be incremented in multiples of 50 Euro Cents.
2. Any lay offers placed by the customer, must be placed in such a way as to ensure that the stake for any corresponding back offer amounts to a minimum of 50 Euro Cents.
3. A placeOrders request may contain up to 50 bet instructions. Any submissions containing more than 50 instructions will fail.
4. We cannot accept betting offers with potential winnings, calculated on the basis of the pre-selected odds that exceed the amount envisaged by article 12, paragraph 4, of Finance Minister Decree no. 111 of 1 March 2006 (10,000 Euros). **N.B. This amount includes the original stake.**
5. placeOrders request containing both back and lay bets in the same order will be rejected.

- How to Access the Italian Exchange API
- Non-interactive Login
- Interactive Login - Desktop Application
- API Login - Desktop Application
- Betting API Endpoints
- Accounts API Endpoints
- Italian Exchange Specific Bet Rules

Betting on Spanish Exchange

How to Access the Spanish Exchange API

Customers who have registered an Spanish Exchange account can access the Betfair Exchange API. Customers can register an account via https://www.betfair.es/prospect/exchange

To use the Spanish Exchange with the API you need to create an Application Key for your Spanish Exchange account by following the process outlined via Application Keys

Once you have created an App Key, you will need to login the API using the Spanish Exchange endpoint which is as follows:

Non-interactive Login

For fully automated applications

```
https://identitysso-cert.betfair.es/api/certlogin
```

Interactive Login - Desktop Application

```
https://identitysso.betfair.es/view/login?product=<AppKey>&url=https://www.betfair.es
```

Interactive Login - API Endpoint

```
https://identitysso.betfair.es/api/login
```

Once a session token has been obtained via the .es login methods above, any further API requests should be sent to the UK Exchange endpoints indicated below.

Requests to these endpoints will automatically return the markets that are available to Spanish Exchange customers

Betting API Endpoints

Interface	Endpoint	JSON-RPC Prefix	<method> Example
JSON-RPC	https://api.betfair.com/exchange/betting/json-rpc/v1	<method>	SportsAPING/v1.0/listMarketBook
JSON REST	https://api.betfair.com/exchange/betting/rest/v1.0/		

Accounts API Endpoints

Interface	Endpoint	JSON-RPC Prefix	<method> Example
JSON-RPC	https://api.betfair.com/exchange/account/json-rpc/v1	<method>	AccountAPING/v1.0/getAccountFunds
JSON REST	https://api.betfair.com/exchange/account/rest/v1.0		

cancelOrders

Operation

cancelOrders

CancelExecutionReport cancelOrders (StringmarketId,List< CancelInstruction >instructions,StringcustomerRef **) throws APINGException**

Cancel all bets OR cancel all bets on a market OR fully or partially cancel particular orders on a market. Only LIMIT orders can be cancelled or partially cancelled once placed.

Parameter name	Type	Required	Description
marketId	String		If marketId **and** betId aren't supplied **all bets** are cancelled. **Please note:** Concurrent requests to cancel all bets will be rejected until the initial request to cancel all bets is complete.
instructions	List< CancelInstruction >		All instructions need to be on the same market. If not supplied all unmatched bets on the market (if market id is passed) are fully cancelled. The limit of cancel instructions per request is 60
customerRef	String		Optional parameter allowing the client to pass a unique string (up to 32 chars) that is used to de-dupe mistaken re-submissions.

Return type	Description
CancelExecutionReport	

Throws	Description
APINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

replaceOrders

Operation

replaceOrders

ReplaceExecutionReport replaceOrders (StringmarketId , List< ReplaceInstruction >instructions ,StringcustomerRef, MarketVersion marketVersion, boolean async **) throws APINGException**

This operation is logically a bulk cancel followed by a bulk place. The cancel is completed first then the new orders are placed. The new orders will be placed atomically in that they will all be placed or none will be placed. In the case where the new orders cannot be placed the cancellations will not be rolled back. See ReplaceInstruction.

Parameter name	Type	Required	Description
marketId	String	✓	The market id these orders are to be placed on
instructions	List< ReplaceInstruction >	✓	The number of replace instructions. The limit of replace instructions per request is 60.
customerRef	String		Optional parameter allowing the client to pass a unique string (up to 32 chars) that is used to de-dupe mistaken re-submissions.
marketVersion	MarketVersion		Optional parameter allowing the client to specify which version of the market the orders should be placed on. If the current market version is higher than that sent on an order, the bet will be lapsed.
async	boolean		An optional flag (not setting equates to false) which specifies if the orders should be replaced asynchronously. Orders can be tracked via the Exchange Stream API or the API-NG by providing a customerOrderRef for each replace order. Not available for MOC or LOC bets.

Return type	Description
ReplaceExecutionReport	

Throws	Description
APINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

updateOrders

Operation

updateOrders

UpdateExecutionReport updateOrders (StringmarketId , List< UpdateInstruction >instructions ,StringcustomerRef **) throws APINGException**

Update non-exposure changing fields

Parameter name	Type	Required	Description
marketId	String	✓	The market id these orders are to be placed on
instructions	List< UpdateInstruction >	✓	The number of update instructions. The limit of update instructions per request is 60
customerRef	String		Optional parameter allowing the client to pass a unique string (up to 32 chars) that is used to de-dupe mistaken re-submissions.

Return type	Description
UpdateExecutionReport	

Throws	Description
APINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

Betting Exceptions

Exceptions

APINGException

This exception is thrown when an operation fails

Error code	Description
TOO_MUCH_DATA	The operation requested too much data, exceeding the Market Data Request Limits.
INVALID_INPUT_DATA	The data input is invalid. A specific description is returned via errorDetails as shown below.
INVALID_SESSION_INFORMATION	The session token hasn't been provided, is invalid or has expired.
NO_APP_KEY	An application key header ('X-Application') has not been provided in the request
NO_SESSION	A session token header ('X-Authentication') has not been provided in the request
UNEXPECTED_ERROR	An unexpected internal error occurred that prevented successful request processing.
INVALID_APP_KEY	The application key passed is invalid or is not present
TOO_MANY_REQUESTS	There are too many pending requests e.g. a listMarketBook with Order/Match projections is limited to 3 concurrent requests. The error also applies to: • listCurrentOrders, listMarketProfitAndLoss and listClearedOrders if you have 3 or more requests currently in execution. • placeOrders, cancelOrders. updateOrders, replaceOrders if the number of transactions submitted exceeds 1000 in a single second.
SERVICE_BUSY	The service is currently too busy to service this request.
TIMEOUT_ERROR	The Internal call to downstream service timed out. **Please note:** If a TIMEOUT error occurs on a placeOrders/replaceOrders request, you should check listCurrentOrders to verify the status of your bets before placing further orders. Please Note: Timeouts will occur after 5 seconds of attempting to process the bet but please allow up to 15 seconds for a timed out order to appear. After this time any unprocessed bets will automatically be Lapsed and no longer be available on the Exchange.
REQUEST_SIZE_EXCEEDS_LIMIT	The request exceeds the request size limit. Requests are limited to a total of 250 betId’s/marketId’s (or a combination of both).
ACCESS_DENIED	The calling client is not permitted to perform the specific action e.g. they have an App Key restriction in place or attempting to place a bet from a restricted jurisdiction.

Other parameters	Type	Required	Description	Values
errorDetails	String		the stack trace of the error	"market id passed is invalid" "locale must use valid iso-639 locale names"

				"currency must use valid iso2 currency code name" "country code must use valid iso2 country code name" "text query has invalid content" "language must use valid iso language name"
requestUUID	String			

Generic JSON-RPC Exceptions

Error Code	Description
-32700	Invalid JSON was received by the server. An error occurred on the server while parsing the JSON text.
-32601	Method not found
-32602	Problem parsing the parameters, or a mandatory parameter was not found
-32603	Internal JSON-RPC error

Betting Enums

Enums

MarketProjection

Value	Description
COMPETITION	If not selected then the competition will not be returned with marketCatalogue
EVENT	If not selected then the event will not be returned with marketCatalogue
EVENT_TYPE	If not selected then the eventType will not be returned with marketCatalogue
MARKET_START_TIME	If not selected then the start time will not be returned with marketCatalogue
MARKET_DESCRIPTION	If not selected then the description will not be returned with marketCatalogue
RUNNER_DESCRIPTION	If not selected then the runners will not be returned with marketCatalogue
RUNNER_METADATA	If not selected then the runner metadata will not be returned with marketCatalogue. If selected then RUNNER_DESCRIPTION will also be returned regardless of whether it is included as a market projection.

PriceData

Value	Description
SP_AVAILABLE	Amount available for the BSP auction.
SP_TRADED	Amount traded in the BSP auction.
EX_BEST_OFFERS	Only the best prices available for each runner, to requested price depth.
EX_ALL_OFFERS	EX_ALL_OFFERS trumps EX_BEST_OFFERS if both settings are present
EX_TRADED	Amount traded on the exchange.

MatchProjection

Value	Description
NO_ROLLUP	No rollup, return raw fragments
ROLLED_UP_BY_PRICE	Rollup matched amounts by distinct matched prices per side.
ROLLED_UP_BY_AVG_PRICE	Rollup matched amounts by average matched price per side

OrderProjection

Value	Description
ALL	EXECUTABLE and EXECUTION_COMPLETE orders
EXECUTABLE	An order that has a remaining unmatched portion. This is either a fully unmatched or partially matched bet (order)
EXECUTION_COMPLETE	An order that does not have any remaining unmatched portion. This is a fully matched bet (order).

MarketStatus

Value	Description
INACTIVE	The market has been created but isn't yet available.
OPEN	The market is open for betting.
SUSPENDED	The market is suspended and not available for betting.
CLOSED	The market has been settled and is no longer available for betting.

RunnerStatus

Value	Description
ACTIVE	ACTIVE
WINNER	WINNER
LOSER	LOSER
PLACED	The runner was placed, applies to EACH_WAY marketTypes only.
REMOVED_VACANT	REMOVED_VACANT applies to Greyhounds. Greyhound markets always return a fixed number of runners (traps). If a dog has been removed, the trap is shown as vacant.
REMOVED	REMOVED
HIDDEN	The selection is hidden from the market. This occurs in Horse Racing markets were runners is hidden when it is doesn't hold an official entry following an entry stage. This could be because the horse was never entered or because they have been scratched from a race at a declaration stage. All matched customer bet prices are set to 1.0 even if there are later supplementary stages. Should it appear likely that a specific runner may actually be supplemented into the race this runner will be reinstated with all matched customer bets set back to the original prices.

TimeGranularity

Value	Description
DAYS	
HOURS	
MINUTES	

Side

Value	Description
BACK	To back a team, horse or outcome is to bet on the selection to win. For LINE markets a Back bet refers to a SELL line. A SELL line will win if the outcome is LESS THAN the taken line (price)
LAY	To lay a team, horse, or outcome is to bet on the selection to lose. For LINE markets a Lay bet refers to a BUY line. A BUY line will win if the outcome is MORE THAN the taken line (price)

OrderStatus

Value	Description
PENDING	An asynchronous order is yet to be processed. Once the bet has been processed by the exchange (including waiting for any in-play delay), the result will be reported and available on the Exchange Stream API and API NG. Not a valid search criteria on MarketFilter
EXECUTION_COMPLETE	An order that does not have any remaining unmatched portion.
EXECUTABLE	An order that has a remaining unmatched portion.
EXPIRED	The order is no longer available for execution due to its time in force constraint. In the case of FILL_OR_KILL orders, this means the order has been killed because it could not be filled to your specifications. Not a valid search criteria on MarketFilter

OrderBy

Value	Description
BY_BET	@Deprecated Use BY_PLACE_TIME instead. Order by placed time, then bet id.
BY_MARKET	Order by market id, then placed time, then bet id.
BY_MATCH_TIME	Order by time of last matched fragment (if any), then placed time, then bet id. Filters out orders which have no matched date. The dateRange filter (if specified) is applied to the matched date.
BY_PLACE_TIME	Order by placed time, then bet id. This is an alias of to be deprecated BY_BET. The dateRange filter (if specified) is applied to the placed date.
BY_SETTLED_TIME	Order by time of last settled fragment (if any due to partial market settlement), then by last match time, then placed time, then bet id. Filters out orders which have not been settled. The dateRange filter (if specified) is applied to the settled date.
BY_VOID_TIME	Order by time of last voided fragment (if any), then by last match time, then placed time, then bet id. Filters out orders which have not been voided. The dateRange filter (if specified) is applied to the voided date.

SortDir

Value	Description
EARLIEST_TO_LATEST	Order from earliest value to latest e.g. lowest betId is first in the results.
LATEST_TO_EARLIEST	Order from the latest value to the earliest e.g. highest betId is first in the results.

OrderType

Value	Description
LIMIT	A normal exchange limit order for immediate execution
LIMIT_ON_CLOSE	Limit order for the auction (SP)
MARKET_ON_CLOSE	Market order for the auction (SP)

MarketSort

Value	Description
MINIMUM_TRADED	Minimum traded volume
MAXIMUM_TRADED	Maximum traded volume
MINIMUM_AVAILABLE	Minimum available to match
MAXIMUM_AVAILABLE	Maximum available to match
FIRST_TO_START	The closest markets based on their expected start time
LAST_TO_START	The most distant markets based on their expected start time

MarketBettingType

Value	Description
ODDS	Odds Market - Any market that doesn't fit any any of the below categories.
LINE	Line Market - LINE markets operate at even-money odds of 2.0. However, price for these markets refers to the line positions available as defined by the markets min-max range and interval steps. Customers either Buy a line (LAY bet, winning if outcome is greater than the taken line (price)) or Sell a line (BACK bet, winning if outcome is less than the taken line (price)). If settled outcome equals the taken line, stake is returned.
RANGE	Range Market - **Now Deprecated**
ASIAN_HANDICAP_DOUBLE_LINE	Asian Handicap Market - A traditional Asian handicap market. Can be identified by marketType ASIAN_HANDICAP
ASIAN_HANDICAP_SINGLE_LINE	Asian Single Line Market - A market in which there can be 0 or multiple winners. e,.g marketType TOTAL_GOALS
FIXED_ODDS	Sportsbook Odds Market. This type is deprecated and will be removed in future releases, when Sportsbook markets will be represented as ODDS market but with a different product type.

ExecutionReportStatus

Value	Description
SUCCESS	Order processed successfully
FAILURE	Order failed.
PROCESSED_WITH_ERRORS	The order itself has been accepted, but at least one (possibly all) actions have generated errors. This error only occurs for **replaceOrders**, **cancelOrders** and **updateOrders** operations. In normal circumstances the placeOrders operation will not return PROCESSED_WITH_ERRORS status as it is an atomic operation. PLEASE NOTE: if the 'Best Execution' features is switched off, placeOrders can return 'PROCESSED_WITH_ERRORS' meaning that some bets can be rejected and other placed when submitted in the same PlaceInstruction
TIMEOUT	The order timed out & the status of the bet is unknown. If a TIMEOUT error occurs on a **placeOrders/replaceOrders** request, you should check **listCurrentOrders** to verify the status of your bets before placing further orders. **Please Note:** Timeouts will occur after 5 seconds of attempting to process the bet but please allow up to 15 seconds for a timed out order to appear. After this time any unprocessed bets will automatically be Lapsed and no longer be available on the Exchange.

ExecutionReportErrorCode

Value	Description
ERROR_IN_MATCHER	The matcher is not healthy. **Please note:** The error will also be returned is you attempt concurrent 'cancel all' bets requests using cancelOrders which isn't permitted.
PROCESSED_WITH_ERRORS	The order itself has been accepted, but at least one (possibly all) actions have generated errors
BET_ACTION_ERROR	There is an error with an action that has caused the entire order to be rejected. Check the instructionReports errorCode for the reason for the rejection of the order.
INVALID_ACCOUNT_STATE	Order rejected due to the account's status (suspended, inactive, dup cards)
INVALID_WALLET_STATUS	Order rejected due to the account's wallet's status
INSUFFICIENT_FUNDS	Account has exceeded its exposure limit or available to bet limit
LOSS_LIMIT_EXCEEDED	The account has exceed the self imposed loss limit
MARKET_SUSPENDED	Market is suspended
MARKET_NOT_OPEN_FOR_BETTING	Market is not open for betting. It is either not yet active, suspended or closed awaiting settlement.
DUPLICATE_TRANSACTION	Duplicate customer reference data submitted - **Please note**: There is a time window associated with the de-duplication of duplicate submissions which is 60 second
INVALID_ORDER	Order cannot be accepted by the matcher due to the combination of actions. For example, bets being edited are not on the same market, or order includes both edits and placement
INVALID_MARKET_ID	Market doesn't exist
PERMISSION_DENIED	Business rules do not allow order to be placed. You are either attempting to place the order using a Delayed Application Key or from a restricted jurisdiction (i.e. USA)
DUPLICATE_BETIDS	duplicate bet ids found
NO_ACTION_REQUIRED	Order hasn't been passed to matcher as system detected there will be no state change
SERVICE_UNAVAILABLE	The requested service is unavailable
REJECTED_BY_REGULATOR	The regulator rejected the order. On the **Italian Exchange** this error will occur if more than 50 bets are sent in a single placeOrders request.
NO_CHASING	A specific error code that relates to Spanish Exchange markets only which indicates that the bet placed contravenes the Spanish regulatory rules relating to loss chasing.
REGULATOR_IS_NOT_AVAILABLE	The underlying regulator service is not available.
TOO_MANY_INSTRUCTIONS	The amount of orders exceeded the maximum amount allowed to be executed
INVALID_MARKET_VERSION	The supplied market version is invalid. Max length allowed for market version is 12.

PersistenceType

Value	Description
LAPSE	Lapse the order when the market is turned in-play
PERSIST	Persist the order to in-play. The bet will be place automatically into the in-play market at the start of the event.
MARKET_ON_CLOSE	Put the order into the auction (SP) at turn-in-play

InstructionReportStatus

Value	Description
SUCCESS	The instruction was successful.
FAILURE	The instruction failed.
TIMEOUT	The order timed out & the status of the bet is unknown. If a TIMEOUT error occurs on a **placeOrders/replaceOrders** request, you should check **listCurrentOrders** to verify the status of your bets before placing further orders. **Please Note:** Timeouts will occur after 5 seconds of attempting to process the bet but please allow up to 15 seconds for a timed out order to appear. After this time any unprocessed bets will automatically be Lapsed and no longer be available on the Exchange.

InstructionReportErrorCode

Value	Description
INVALID_BET_SIZE	bet size is invalid for your currency or your regulator
INVALID_RUNNER	Runner does not exist, includes vacant traps in greyhound racing
BET_TAKEN_OR_LAPSED	Bet cannot be cancelled or modified as it has already been taken or has been cancelled/lapsed Includes attempts to cancel /modify market on close BSP bets and cancelling limit on close BSP bets. The error may be returned on placeOrders request if for example a bet is placed at the point when a market admin event takes place (i.e. market is turned in-play)
BET_IN_PROGRESS	No result was received from the matcher in a timeout configured for the system
RUNNER_REMOVED	Runner has been removed from the event
MARKET_NOT_OPEN_FOR_BETTING	Attempt to edit a bet on a market that has closed.
LOSS_LIMIT_EXCEEDED	The action has caused the account to exceed the self imposed loss limit
MARKET_NOT_OPEN_FOR_BSP_BETTING	Market now closed to bsp betting. Turned in-play or has been reconciled
INVALID_PRICE_EDIT	Attempt to edit down the price of a bsp limit on close lay bet, or edit up the price of a limit on close back bet
INVALID_ODDS	Odds not on price ladder - either edit or placement
INSUFFICIENT_FUNDS	Insufficient funds available to cover the bet action. Either the exposure limit or available to bet limit would be exceeded
INVALID_PERSISTENCE_TYPE	Invalid persistence type for this market, e.g. KEEP for a non in-play market.
ERROR_IN_MATCHER	A problem with the matcher prevented this action completing successfully
INVALID_BACK_LAY_COMBINATION	The order contains a back and a lay for the same runner at overlapping prices. This would guarantee a self match. This also applies to BSP limit on close bets
ERROR_IN_ORDER	The action failed because the parent order failed
INVALID_BID_TYPE	Bid type is mandatory
INVALID_BET_ID	Bet for id supplied has not been found
	Bet cancelled but replacement bet was not placed

CANCELLED_NOT_PLACED	
RELATED_ACTION_FAILED	Action failed due to the failure of a action on which this action is dependent
NO_ACTION_REQUIRED	the action does not result in any state change. eg changing a persistence to it's current value
TIME_IN_FORCE_CONFLICT	You may only specify a time in force on either the place request OR on individual limit order instructions (not both), since the implied behaviors are incompatible.
UNEXPECTED_PERSISTENCE_TYPE	You have specified a persistence type for a FILL_OR_KILL order, which is nonsensical because no umatched portion can remain after the order has been placed.
INVALID_ORDER_TYPE	You have specified a time in force of FILL_OR_KILL, but have included a non-LIMIT order type.
UNEXPECTED_MIN_FILL_SIZE	You have specified a minFillSize on a limit order, where the limit order's time in force is not FILL_OR_KILL. Using minFillSize is not supported where the time in force of the request (as opposed to an order) is FILL_OR_KILL.
INVALID_CUSTOMER_ORDER_REF	The supplied customer order reference is too long.
INVALID_MIN_FILL_SIZE	The minFillSize must be greater than zero and less than or equal to the order's size. The minFillSize cannot be less than the minimum bet size for your currency
BET_LAPSED_PRICE_IMPROVEMENT_TOO_LARGE	Your bet is lapsed. There is better odds than requested available in the market, but your preferences don't allow the system to match your bet against better odds. Change your betting preferences to accept better odds if you don't want to receive this error.

RollupModel

Value	Description
STAKE	The volumes will be rolled up to the minimum value which is >= rollupLimit.
PAYOUT	The volumes will be rolled up to the minimum value where the payout(price * volume) is >= rollupLimit. On a LINE market, volumes will be rolled up where payout(2.0 * volume) is >= rollupLimit
MANAGED_LIABILITY	The volumes will be rolled up to the minimum value which is >= rollupLimit, until a lay price threshold. There after, the volumes will be rolled up to the minimum value such that the liability >= a minimum liability. Not supported as yet.
NONE	No rollup will be applied. However the volumes will be filtered by currency specific minimum stake unless overridden specifically for the channel.

GroupBy

Value	Description
EVENT_TYPE	A roll up of settled P&L, commission paid and number of bet orders, on a specified event type
EVENT	A roll up of settled P&L, commission paid and number of bet orders, on a specified event
MARKET	A roll up of settled P&L, commission paid and number of bet orders, on a specified market
SIDE	An averaged roll up of settled P&L, and number of bets, on the specified side of a specified selection within a specified market, that are either settled or voided
BET	The P&L, side and regulatory information etc, about each individual bet order.

BetStatus

Value	Description
SETTLED	A matched bet that was settled normally
VOIDED	A matched bet that was subsequently voided by Betfair, before, during or after settlement
LAPSED	Unmatched bet that was cancelled by Betfair (for example at turn in play).
CANCELLED	Unmatched bet that was cancelled by an explicit customer action.

marketType - Legacy Data

Value	Description
A	Asian Handicap
L	Line market
O	Odds market
R	Range market.
NOT_APPLICABLE	The market does not have an applicable marketType.

TimeInForce

Value	Description
FILL_OR_KILL	Execute the transaction immediately and completely (filled to size or between minFillSize and size) or not at all (cancelled). For LINE markets Volume Weighted Average Price (VWAP) functionality is disabled

BetTargetType

Value	Description
BACKERS_PROFIT	The payout requested minus the calculated size at which this LimitOrder is to be placed. BetTargetType bets are invalid for LINE markets
PAYOUT	The total payout requested on a LimitOrder

PriceLadderType

Value	Description
CLASSIC	Price ladder increments traditionally used for Odds Markets.
FINEST	Price ladder with the finest available increment, traditionally used for Asian Handicap markets.
LINE_RANGE	Price ladder used for LINE markets. Refer to MarketLineRangeInfo for more details.

Betting Type Definitions

Type definitions

MarketFilter

Field name	Type	Required	Description
textQuery	String		Restrict markets by any text associated with the Event name. You can include a wildcard (*) character as long as it is not the first character. **Please note -** the textQuery field doesn't evaluate market or selection names.
exchangeIds	Set<String>		**DEPRECATED**
eventTypeIds	Set<String>		Restrict markets by event type associated with the market. (i.e., Football, Hockey, etc)
eventIds	Set<String>		Restrict markets by the event id associated with the market.
competitionIds	Set<String>		Restrict markets by the competitions associated with the market.
marketIds	Set<String>		Restrict markets by the market id associated with the market.
venues	Set<String>		Restrict markets by the venue associated with the market. Currently only Horse Racing markets have venues.
bspOnly	boolean		Restrict to bsp markets only, if True or non-bsp markets if False. If not specified then returns both BSP and non-BSP markets
turnInPlayEnabled	boolean		Restrict to markets that will turn in play if True or will not turn in play if false. If not specified, returns both.
inPlayOnly	boolean		Restrict to markets that are currently in play if True or are not currently in play if false. If not specified, returns both.
marketBettingTypes	Set< MarketBettingType >		Restrict to markets that match the betting type of the market (i.e. Odds, Asian Handicap Singles, Asian Handicap Doubles or Line)
marketCountries	Set<String>		Restrict to markets that are in the specified country or countries
marketTypeCodes	Set<String>		Restrict to markets that match the type of the market (i.e., MATCH_ODDS, HALF_TIME_SCORE). You should use this instead of relying on the market name as the market type codes are the same in all locales. **Please note:** All marketTypes are available via the listMarketTypes operations.
marketStartTime	TimeRange		Restrict to markets with a market start time before or after the specified date
withOrders	Set< OrderStatus >		Restrict to markets that I have one or more orders in these status.
raceTypes	Set<String>		Restrict by race type (i.e. Hurdle, Flat, Bumper, Harness, Chase)

MarketCatalogue

Information about a market

Field name	Type	Required	Description
marketId	String	✓	The unique identifier for the market. MarketId's are prefixed with '1.'
marketName	String	✓	The name of the market
marketStartTime	Date		The time this market starts at, only returned when the MARKET_START_TIME enum is passed in the marketProjections
description	MarketDescription		Details about the market
totalMatched	Double		The total amount of money matched on the market
runners	List< RunnerCatalog >		The runners (selections) contained in the market
eventType	EventType		The Event Type the market is contained within

competition	Competition		The competition the market is contained within. Usually only applies to Football competitions
event	Event		The event the market is contained within

MarketBook

The dynamic data in a market

Field name	Type	Required	Description
marketId	String	✓	The unique identifier for the market. MarketId's are prefixed with '1.'
isMarketDataDelayed	boolean	✓	True if the data returned by listMarketBook will be delayed. The data may be delayed because you are not logged in with a funded account or you are using an Application Key that does not allow up to date data.
status	MarketStatus		The status of the market, for example OPEN, SUSPENDED, CLOSED (settled), etc.
betDelay	int		The number of seconds an order is held until it is submitted into the market. Orders are usually delayed when the market is in-play
bspReconciled	boolean		True if the market starting price has been reconciled
complete	boolean		If false, runners may be added to the market
inplay	boolean		True if the market is currently in play
numberOfWinners	int		The number of selections that could be settled as winners
numberOfRunners	int		The number of runners in the market
numberOfActiveRunners	int		The number of runners that are currently active. An active runner is a selection available for betting
lastMatchTime	Date		The most recent time an order was executed
totalMatched	double		The total amount matched
totalAvailable	double		The total amount of orders that remain unmatched
crossMatching	boolean		True if cross matching is enabled for this market.
runnersVoidable	boolean		True if runners in the market can be voided. **Please note** - this doesn't include horse racing markets under which bets are voided on non-runners with any applicable reduction factor applied/
version	long		The version of the market. The version increments whenever the market status changes, for example, turning in-play, or suspended when a goal is scored.
runners	List< Runner >		Information about the runners (selections) in the market.
keyLineDescription	KeyLineDescription		Description of a markets key line for valid market types

RunnerCatalog

Information about the Runners (selections) in a market

Field name	Type	Required	Description
selectionId	long	✓	The unique id for the selection. **Please note:** The selectionId can be mapped to the runner name using the output from **listMarketCatalogue**
runnerName	String	✓	The name of the runner
handicap	double	✓	The handicap applies to market with the MarketBettingType ASIAN_HANDICAP_SINGLE_LINE & ASIAN_HANDICAP_DOUBLE_LINE only otherwise '0'
sortPriority	int	✓	The sort priority of this runner. Indicates the order in which the runners are displayed on the Betfair Exchange website.
metadata	Map<String, String>		Metadata associated with the runner. For a description of this data for Horse Racing, please see Runner Metadata Description

Runner

The dynamic data about runners in a market

Field name	Type	Required	Description
selectionId	long	✓	The unique id of the runner (selection)
handicap	double	✓	The handicap. Enter the specific handicap value (returned by RUNNER in listMaketBook) if the market is an Asian handicap market.
status	RunnerStatus	✓	The status of the selection (i.e., ACTIVE, REMOVED, WINNER, PLACED, LOSER, HIDDEN) Runner status information is available for 90 days following market settlement.
adjustmentFactor	double	✓	The adjustment factor applied if the selection is removed
lastPriceTraded	double		The price of the most recent bet matched on this selection
totalMatched	double		The total amount matched on this runner
removalDate	Date		If date and time the runner was removed
sp	StartingPrices		The BSP related prices for this runner
ex	ExchangePrices		The Exchange prices available for this runner
orders	List< Order >		List of orders in the market
matches	List< Match >		List of matches (i.e, orders that have been fully or partially executed)
matchesByStrategy	Map<String,Matches>		List of matches for each strategy, ordered by matched data

StartingPrices

Information about the Betfair Starting Price. Only available in BSP markets

Field name	Type	Required	Description
nearPrice	double		What the starting price would be if the market was reconciled now taking into account the SP bets as well as unmatched exchange bets on the same selection in the exchange. This data is cached and update every 60 seconds. **Please note:** Type Double may contain numbers, INF, -INF, and NaN.
farPrice	double		What the starting price would be if the market was reconciled now taking into account only the currently place SP bets. The Far Price is not as complicated but not as accurate and only accounts for money on the exchange at SP. This data is cached and updated every 60 seconds. **Please note:** Type Double may contain numbers, INF, -INF, and NaN.
backStakeTaken	List< PriceSize >		The total amount of back bets matched at the actual Betfair Starting Price. Pre-reconciliation, this field is zero for all prices except 1.01 (for Market on Close bets) and at the limit price for any Limit on Close bets.
layLiabilityTaken	List< PriceSize >		The lay amount matched at the actual Betfair Starting Price. Pre-reconciliation, this field is zero for all prices except 1000 (for Market on Close bets) and at the limit price for any Limit on Close bets.
actualSP	double		The final BSP price for this runner. Only available for a BSP market that has been reconciled.

ExchangePrices

Field name	Type	Required	Description
availableToBack	List< PriceSize >		
availableToLay	List< PriceSize >		
tradedVolume	List< PriceSize >		

Event

Event

Field name	Type	Required	Description
id	String		The unique id for the event
name	String		The name of the event
countryCode	String		The ISO-2 code for the event. A list of ISO-2 codes is available via http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2
timezone	String		This is timezone in which the event is taking place.
venue	String		venue
openDate	Date		The scheduled start date and time of the event. This is Europe/London (GMT) by default

EventResult

Event Result

Field name	Type	Required	Description
event	Event		Event
marketCount	int		Count of markets associated with this event

Competition

Competition

Field name	Type	Required	Description
id	String		id
name	String		name

CompetitionResult

Competition Result

Field name	Type	Required	Description
competition	Competition		Competition
marketCount	int		Count of markets associated with this competition
competitionRegion	String		Region in which this competition is happening

EventType

EventType

Field name	Type	Required	Description
id	String		id
name	String		name

EventTypeResult

EventType Result

Field name	Type	Required	Description
eventType	EventType		The ID identifying the Event Type
marketCount	int		Count of markets associated with this eventType

MarketTypeResult

MarketType Result

Field name	Type	Required	Description
marketType	String		Market Type
marketCount	int		Count of markets associated with this marketType

CountryCodeResult

CountryCode Result

Field name	Type	Required	Description
countryCode	String		The ISO-2 code for the event. A list of ISO-2 codes is available via http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2
marketCount	int		Count of markets associated with this Country Code

VenueResult

Venue Result

Field name	Type	Required	Description
venue	String		Venue
marketCount	int		Count of markets associated with this Venue

TimeRange

TimeRange

Field name	Type	Required	Description
from	Date		from
to	Date		to

TimeRangeResult

TimeRange Result

Field name	Type	Required	Description
timeRange	TimeRange		TimeRange
marketCount	int		Count of markets associated with this TimeRange

Order

Field name	Type	Required	Description
betId	String	✓	
orderType	OrderType	✓	BSP Order type.
status	OrderStatus	✓	Either EXECUTABLE (an unmatched amount remains) or EXECUTION_COMPLETE (no unmatched amount remains).
persistenceType	PersistenceType	✓	What to do with the order at turn-in-play
side	Side	✓	Indicates if the bet is a Back or a LAY. For LINE markets customers either Buy a line (LAY bet, winning if outcome is greater than the taken line (price)) or Sell a line (BACK bet, winning if outcome is less than the taken line (price))
price	double	✓	The price of the bet. **Please note**: LINE markets operate at even-money odds of 2.0. However, price for these markets refers to the line positions available as defined by the markets min-max range and interval steps
size	double	✓	The size of the bet.
bspLiability	double	✓	Not to be confused with size. This is the liability of a given BSP bet.
placedDate	Date	✓	The date, to the second, the bet was placed.
avgPriceMatched	double		The average price matched at. Voided match fragments are removed from this average calculation. For MARKET_ON_CLOSE BSP bets this reports the matched SP price following the SP reconciliation process. This value is not meaningful for activity on LINE markets and is not guaranteed to be returned or maintained for these markets.
sizeMatched	double		The current amount of this bet that was matched.
sizeRemaining	double		The current amount of this bet that is unmatched.
sizeLapsed	double		The current amount of this bet that was lapsed.
sizeCancelled	double		The current amount of this bet that was cancelled.
sizeVoided	double		The current amount of this bet that was voided.
customerOrderRef	CustomerOrderRef		The customer order reference sent for this bet
customerStrategyRef	CustomerStrategyRef		The customer strategy reference sent for this bet

Match

An individual bet Match, or rollup by price or avg price. Rollup depends on the requested MatchProjection

Field name	Type	Required	Description
betId	String		Only present if no rollup
matchId	String		Only present if no rollup
side	Side	✓	Indicates if the bet is a Back or a LAY
price	double	✓	Either actual match price or avg match price depending on rollup. This value is not meaningful for activity on LINE markets and is not guaranteed to be returned or maintained for these markets.

size	double	✓	Size matched at in this fragment, or at this price or avg price depending on rollup
matchDate	Date		Only present if no rollup

MarketVersion

Market version

Field name	Type	Required	Description
version	long		A non-monotonically increasing number indicating market changes

MarketDescription

Market definition

Field name	Type	Required	Description
persistenceEnabled	boolean	✓	If 'true' the market supports 'Keep' bets if the market is to be turned in-play
bspMarket	boolean	✓	If 'true' the market supports Betfair SP betting
marketTime	Date	✓	The market start time
suspendTime	Date	✓	The market suspend time
settleTime	Date		settled time
bettingType	MarketBettingType	✓	See MarketBettingType
turnInPlayEnabled	boolean	✓	If 'true' the market is set to turn in-play
marketType	String	✓	Market base type
regulator	String	✓	The market regulator. Value include “GIBRALTAR REGULATOR" (.com), MR_ESP (Betfair.es markets), MR_IT (Betfair.it). GIBRALTAR REGULATOR = MR_INT in the Stream API
marketBaseRate	double	✓	The commission rate applicable to the market
discountAllowed	boolean	✓	Indicates whether or not the user's discount rate is taken into account on this market. If ‘false’ all users will be charged the same commission rate, regardless of discount rate.
wallet	String		The wallet to which the market belongs.
rules	String		The market rules.
rulesHasDate	boolean		Indicates whether rules have a date included.
eachWayDivisor	double		The divisor is returned for the marketType EACH_WAY only and refers to the fraction of the win odds at which the place portion of an each way bet is settled
clarifications	String		Any additional information regarding the market
lineRangeInfo	MarketLineRangeInfo		Line range info for line markets
raceType	String		An external identifier of a race type
priceLadderDescription	PriceLadderDescription		Details about the price ladder in use for this market.

MarketRates

Market Rates

Field name	Type	Required	Description

marketBaseRate	double	✓	marketBaseRate
discountAllowed	boolean	✓	discountAllowed

MarketLicence

Market Licence

Field name	Type	Required	Description
wallet	String	✓	The wallet from which funds will be taken when betting on this market
rules	String		The rules for this market
rulesHasDate	boolean		The market's start date and time are relevant to the rules.
clarifications	String		Clarifications to the rules for the market

MarketLineRangeInfo

Market Line and Range Info

Field name	Type	Required	Description
maxUnitValue	double	✓	maxPrice - Maximum value for the outcome, in market units for this market (eg 100 runs)
minUnitValue	double	✓	minPrice - Minimum value for the outcome, in market units for this market (eg 0 runs)
interval	double	✓	interval - The odds ladder on this market will be between the range of minUnitValue and maxUnitValue, in increments of the inverval value.e.g. If minUnitValue=10 runs, maxUnitValue=20 runs, interval=0.5 runs, then valid odds include 10, 10.5, 11, 11.5 up to 20 runs.
marketUnit	String	✓	unit - The type of unit the lines are incremented in by the interval (e.g: runs, goals or seconds.

PriceSize

Field name	Type	Required	Description
price	double	✓	The price available
size	double	✓	The stake available

ClearedOrderSummary

Summary of a cleared order.

Field name	Type	Required	Description
eventTypeId	EventTypeId		The id of the event type bet on. Available at EVENT_TYPE groupBy level or lower.
eventId	EventId		The id of the event bet on. Available at EVENT groupBy level or lower.
marketId	MarketId		The id of the market bet on. Available at MARKET groupBy level or lower.
selectionId	SelectionId		The id of the selection bet on. Available at RUNNER groupBy level or lower.
handicap	Handicap		The handicap. Enter the specific handicap value (returned by RUNNER in listMaketBook) if the market is an Asian handicap market. Available at MARKET groupBy level or lower.

betId	BetId		The id of the bet. Available at BET groupBy level.
placedDate	Date		The date the bet order was placed by the customer. Only available at BET groupBy level.
persistenceType	PersistenceType		The turn in play persistence state of the order at bet placement time. This field will be empty or omitted on true SP bets. Only available at BET groupBy level.
orderType	OrderType		The type of bet (e.g standard limited-liability Exchange bet (LIMIT), a standard BSP bet (MARKET_ON_CLOSE), or a minimum-accepted-price BSP bet (LIMIT_ON_CLOSE)). If the bet has a OrderType of MARKET_ON_CLOSE and a persistenceType of MARKET_ON_CLOSE then it is a bet which has transitioned from LIMIT to MARKET_ON_CLOSE. Only available at BET groupBy level.
side	Side		Whether the bet was a back or lay bet. Available at SIDE groupBy level or lower.
itemDescription	ItemDescription		A container for all the ancillary data and localised text valid for this Item
betOutcome	String		The settlement outcome of the bet. Tri-state (WIN/LOSE/PLACE) to account for Each Way bets where the place portion of the bet won but the win portion lost. The profit/loss amount in this case could be positive or negative depending on the price matched at. Only available at BET groupBy level.
priceRequested	Price		The average requested price across all settled bet orders under this Item. Available at SIDE groupBy level or lower. For LINE markets this is the line position requested. For LINE markets this is the line position requested.
settledDate	Date		The date and time the bet order was settled by Betfair. Available at SIDE groupBy level or lower.
lastMatchedDate	Date		The date and time the last bet order was matched by Betfair. Available on Settled orders only.
betCount	int		The number of actual bets within this grouping (will be 1 for BET groupBy)
commission	Size		The cumulative amount of commission paid by the customer across all bets under this Item, in the account currency. Available at EXCHANGE, EVENT_TYPE, EVENT and MARKET level groupings only.
priceMatched	Price		The average matched price across all settled bets or bet fragments under this Item. Available at SIDE groupBy level or lower. For LINE markets this is the line position matched at.
priceReduced	boolean		If true, then the matched price was affected by a reduction factor due to of a runner removal from this Horse Racing market.
sizeSettled	Size		The cumulative bet size that was settled as matched or voided under this Item, in the account currency. Available at SIDE groupBy level or lower.
profit	Size		The profit or loss (negative profit) gained on this line, in the account currency
sizeCancelled	Size		The amount of the bet that was available to be matched, before cancellation or lapsing, in the account currency
customerOrderRef	String		The order reference defined by the customer for the bet order
customerStrategyRef	String		The strategy reference defined by the customer for the bet order

ClearedOrderSummaryReport

A container representing search results.

Field name	Type	Required	Description
clearedOrders	List<ClearedOrderSummary>	✓	The list of cleared orders returned by your query. This will be a valid list (i.e. empty or non-empty but never 'null').
moreAvailable	boolean	✓	Indicates whether there are further result items beyond this page. Note that underlying data is highly time-dependent and the subsequent search orders query might return an empty result.

ItemDescription

This object contains some text which may be useful to render a betting history view. It offers no long-term warranty as to the correctness of the text.

	Type	Required	Description

Field name			
eventTypeDesc	String		The event type name, translated into the requested locale. Available at EVENT_TYPE groupBy or lower.
eventDesc	String		The eventName, or openDate + venue, translated into the requested locale. Available at EVENT groupBy or lower.
marketDesc	String		The market name or racing market type ("Win", "To Be Placed (2 places)", "To Be Placed (5 places)" etc) translated into the requested locale. Available at MARKET groupBy or lower.
marketType	String		The market type e.g. MATCH_ODDS, PLACE, WIN etc.
marketStartTime	Date		The start time of the market (in ISO-8601 format, not translated). Available at MARKET groupBy or lower.
runnerDesc	String		The runner name, maybe including the handicap, translated into the requested locale. Available at BET groupBy.
numberOfWinners	int		The number of winners on a market. Available at BET groupBy.
eachWayDivisor	double		The divisor is returned for the marketType EACH_WAY only and refers to the fraction of the win odds at which the place portion of an each way bet is settled

RunnerId

This object contains the unique identifier for a runner

Field name	Type	Required	Description
marketId	MarketId	✓	The id of the market bet on
selectionId	SelectionId	✓	The id of the selection bet on
handicap	Handicap		The handicap associated with the runner in case of asian handicap markets, otherwise returns '0.0'.

CurrentOrderSummaryReport

A container representing search results.

Field name	Type	Required	Description
currentOrders	List< CurrentOrderSummary >	✓	The list of current orders returned by your query. This will be a valid list (i.e. empty or non-empty but never 'null').
moreAvailable	boolean	✓	Indicates whether there are further result items beyond this page. Note that underlying data is highly time-dependent and the subsequent search orders query might return an empty result.

CurrentOrderSummary

Summary of a current order.

Field name	Type	Required	Description
betId	String	✓	The bet ID of the original place order.
marketId	String	✓	The market id the order is for.
selectionId	long	✓	The selection id the order is for.
handicap	double	✓	The handicap associated with the runner in case of Asian handicap markets, null otherwise.
priceSize	PriceSize	✓	The price and size of the bet.
bspLiability	double	✓	Not to be confused with size. This is the liability of a given BSP bet.

side	Side	✓	BACK/LAY
status	OrderStatus	✓	Either EXECUTABLE (an unmatched amount remains) or EXECUTION_COMPLETE (no unmatched amount remains).
persistenceType	PersistenceType	✓	What to do with the order at turn-in-play.
orderType	OrderType	✓	BSP Order type.
placedDate	Date	✓	The date, to the second, the bet was placed.
matchedDate	Date	✓	The date, to the second, of the last matched bet fragment (where applicable)
averagePriceMatched	double		The average price matched at. Voided match fragments are removed from this average calculation. The price is automatically adjusted in the event of non runners being declared with applicable reduction factors. **Please note:** This value is not meaningful for activity on LINE markets and is not guaranteed to be returned or maintained for these markets.
sizeMatched	double		The current amount of this bet that was matched.
sizeRemaining	double		The current amount of this bet that is unmatched.
sizeLapsed	double		The current amount of this bet that was lapsed.
sizeCancelled	double		The current amount of this bet that was cancelled.
sizeVoided	double		The current amount of this bet that was voided.
regulatorAuthCode	String		The regulator authorisation code.
regulatorCode	String		The regulator Code.
customerOrderRef	String		The order reference defined by the customer for this bet
customerStrategyRef	String		The strategy reference defined by the customer for this bet

PlaceInstruction

Instruction to place a new order

Field name	Type	Required	Description
orderType	OrderType	✓	
selectionId	long	✓	The selection_id.
handicap	double		The handicap associated with the runner in case of Asian handicap markets (e.g. marketTypes ASIAN_HANDICAP_DOUBLE_LINE, ASIAN_HANDICAP_SINGLE_LINE) null otherwise.
side	Side	✓	Back or Lay
limitOrder	LimitOrder		A simple exchange bet for immediate execution
limitOnCloseOrder	LimitOnCloseOrder		Bets are matched if, and only if, the returned starting price is better than a specified price. In the case of back bets, LOC bets are matched if the calculated starting price is greater than the specified price. In the case of lay bets, LOC bets are matched if the starting price is less than the specified price. If the specified limit is equal to the starting price, then it may be matched, partially matched, or may not be matched at all, depending on how much is needed to balance all bets against each other (MOC, LOC and normal exchange bets)
marketOnCloseOrder	MarketOnCloseOrder		Bets remain unmatched until the market is reconciled. They are matched and settled at a price that is representative of the market at the point the market is turned in-play. The market is reconciled to find a starting price and MOC bets are settled at whatever starting price is returned. MOC bets are always matched and settled, unless a starting price is not available for the selection. Market on Close bets can only be placed before the starting price is determined
	String		An optional reference customers can set to identify instructions.. No validation will be done on uniqueness and the string is limited to 32 characters. If an empty string is provided it will be treated as null.

customerOrderRef			

PlaceExecutionReport

Field name	Type	Required	Description
customerRef	String		Echo of the customerRef if passed.
status	ExecutionReportStatus	✓	
errorCode	ExecutionReportErrorCode		
marketId	String		Echo of marketId passed
instructionReports	List< PlaceInstructionReport >		

LimitOrder

Place a new LIMIT order (simple exchange bet for immediate execution)

Field name	Type	Required	Description
size	double	✓	The size of the bet. **Please note**: For market type EACH_WAY. The total stake = size x 2
price	double	✓	The limit price. For LINE markets, the price at which the bet is settled and struck will always be 2.0 (Evens). On these bets, the Price field is used to indicate the line value which is being bought or sold
persistenceType	PersistenceType	✓	What to do with the order at turn-in-play
timeInForce	TimeInForce		The type of TimeInForce value to use. This value takes precedence over any PersistenceType value chosen. If this attribute is populated along with the PersistenceType field, then the PersistenceType will be ignored. When using FILL_OR_KILL for a Line market the Volume Weighted Average Price (VWAP) functionality is disabled
minFillSize	Size		An optional field used if the TimeInForce attribute is populated. If specified without TimeInForce then this field is ignored. If no minFillSize is specified, the order is killed unless the entire size can be matched. If minFillSize is specified, the order is killed unless at least the minFillSize can be matched. The minFillSize cannot be greater than the order's size. If specified for a BetTargetType and FILL_OR_KILL order, then this value will be ignored
betTargetType	BetTargetType		An optional field to allow betting to a targeted PAYOUT or BACKERS_PROFIT. It's invalid to specify both a Size and BetTargetType Matching provides best execution at the requested price or better up to the payout or profit. If the bet is not matched completely and immediately, the remaining portion enters the unmatched pool of bets on the exchange BetTargetType bets are invalid for LINE markets
betTargetSize	Size		An optional field which must be specified if BetTargetType is specified for this order The requested outcome size of either the payout or profit. This is named from the backer's perspective. For Lay bets the profit represents the bet's liability

LimitOnCloseOrder

Place a new LIMIT_ON_CLOSE bet

Field name	Type	Required	Description
liability	double	✓	The size of the bet. See Min BSP Liability
price	double	✓	The limit price of the bet if LOC

MarketOnCloseOrder

Place a new MARKET_ON_CLOSE bet

Field name	Type	Required	Description
liability	double	✓	The size of the bet.

PlaceInstructionReport

Response to a PlaceInstruction

Field name	Type	Required	Description
status	InstructionReportStatus	✓	whether the command succeeded or failed
errorCode	InstructionReportErrorCode		cause of failure, or null if command succeeds
orderStatus	OrderStatus		The status of the order, if the instruction succeeded. If the instruction was unsuccessful, no value is provided.
instruction	PlaceInstruction	✓	The instruction that was requested
betId	String		The bet ID of the new bet. Will be null on failure or if order was placed asynchronously.
placedDate	Date		Will be null if order was placed asynchronously
averagePriceMatched	Price		Will be null if order was placed asynchronously. This value is not meaningful for activity on LINE markets and is not guaranteed to be returned or maintained for these markets.
sizeMatched	Size		Will be null if order was placed asynchronously

CancelInstruction

Instruction to fully or partially cancel an order (only applies to LIMIT orders). **Please note:** the CancelInstruction report won't be returned for marketId level cancel instructions.

Field name	Type	Required	Description
betId	String		The betId
sizeReduction	double		If supplied then this is a partial cancel. Should be set to 'null' if no size reduction is required.

CancelExecutionReport

Field name	Type	Required	Description
customerRef	String		Echo of the customerRef if passed.
status	ExecutionReportStatus	✓	
errorCode	ExecutionReportErrorCode		
marketId	String		Echo of marketId passed
instructionReports	List< CancelInstructionReport >		

ReplaceInstruction

Instruction to replace a LIMIT or LIMIT_ON_CLOSE order at a new price. Original order will be cancelled and a new order placed at the new price for the remaining stake.

Field name	Type	Required	Description
betId	String	✓	Unique identifier for the bet
newPrice	double	✓	The price to replace the bet at

ReplaceExecutionReport

Field name	Type	Required	Description
customerRef	String		Echo of the customerRef if passed.
status	ExecutionReportStatus	✓	
errorCode	ExecutionReportErrorCode		
marketId	String		Echo of marketId passed
instructionReports	List< ReplaceInstructionReport >		

ReplaceInstructionReport

Field name	Type	Required	Description
status	InstructionReportStatus	✓	whether the command succeeded or failed
errorCode	InstructionReportErrorCode		cause of failure, or null if command succeeds
cancelInstructionReport	CancelInstructionReport		Cancelation report for the original order
placeInstructionReport	PlaceInstructionReport		Placement report for the new order

CancelInstructionReport

Field name	Type	Required	Description
status	InstructionReportStatus	✓	whether the command succeeded or failed
errorCode	InstructionReportErrorCode		cause of failure, or null if command succeeds
instruction	CancelInstruction		The instruction that was requested
sizeCancelled	double	✓	
cancelledDate	Date		

UpdateInstruction

Instruction to update LIMIT bet's persistence of an order that do not affect exposure

Field name	Type	Required	Description
betId	String	✓	Unique identifier for the bet
newPersistenceType	PersistenceType	✓	The new persistence type to update this bet to

UpdateExecutionReport

Field name	Type	Required	Description
customerRef	String		Echo of the customerRef if passed.
status	ExecutionReportStatus	✓	
errorCode	ExecutionReportErrorCode		
marketId	String		Echo of marketId passed
instructionReports	List< UpdateInstructionReport >		

UpdateInstructionReport

Field name	Type	Required	Description
status	InstructionReportStatus	✓	whether the command succeeded or failed
errorCode	InstructionReportErrorCode		cause of failure, or null if command succeeds
instruction	UpdateInstruction	✓	The instruction that was requested

Type	Required	Description

PriceProjection

Selection criteria of the returning price data

Field name	Type	Required	Description
priceData	Set< PriceData >		The basic price data you want to receive in the response.
exBestOffersOverrides	ExBestOffersOverrides		Options to alter the default representation of best offer prices Applicable to EX_BEST_OFFERS priceData selection
virtualise	boolean		Indicates if the returned prices should include virtual prices. Applicable to EX_BEST_OFFERS and EX_ALL_OFFERS priceData selections, default value is false. **Please note:** This must be set to 'true' replicate the display of prices on the Betfair Exchange website.
rolloverStakes	boolean		Indicates if the volume returned at each price point should be the absolute value or a cumulative sum of volumes available at the price and all better prices. If unspecified defaults to false. Applicable to EX_BEST_OFFERS and EX_ALL_OFFERS price projections. Not supported as yet.

ExBestOffersOverrides

Options to alter the default representation of best offer prices

Field name	Type	Required	Description
bestPricesDepth	int		The maximum number of prices to return on each side for each runner. If unspecified defaults to 3. Maximum returned price depth returned is 10.
rollupModel	RollupModel		The model to use when rolling up available sizes. If unspecified defaults to STAKE rollup model with rollupLimit of minimum stake in the specified currency.
rollupLimit	int		The volume limit to use when rolling up returned sizes. The exact definition of the limit depends on the rollupModel. If no limit is provided it will use minimum stake as default the value. Ignored if no rollup model is specified.
rollupLiabilityThreshold	double		Only applicable when rollupModel is MANAGED_LIABILITY. The rollup model switches from being stake based to liability based at the smallest lay price which is >= rollupLiabilityThreshold.service level default (TBD). Not supported as yet.
rollupLiabilityFactor	int		Only applicable when rollupModel is MANAGED_LIABILITY. (rollupLiabilityFactor * rollupLimit) is the minimum liabilty the user is deemed to be comfortable with. After the rollupLiabilityThreshold price subsequent volumes will be rolled up to minimum value such that the liability >= the minimum liability.service level default (5). Not supported as yet.

MarketProfitAndLoss

Profit and loss in a market

Field name	Type	Required	Description
marketId	String		The unique identifier for the market
commissionApplied	double		The commission rate applied to P&L values. Only returned if netOfCommision option is requested
profitAndLosses	List<RunnerProfitAndLoss>		Calculated profit and loss data.

RunnerProfitAndLoss

Profit and loss if selection is wins or loses

Field name	Type	Required	Description
selectionId	SelectionId		The unique identifier for the selection
ifWin	double		Profit or loss for the market if this selection is the winner.
ifLose	double		Profit or loss for the market if this selection is the loser. Only returned for multi-winner odds markets.
ifPlace	double		Profit or loss for the market if this selection is placed. Applies to marketType EACH_WAY only.

PriceLadderDescription

Description of the price ladder type and any related data.

Field name	Type	Required	Description
type	PriceLadderType	✓	The type of price ladder.

KeyLineSelection

Description of a markets key line selection, comprising the selectionId and handicap of the team it is applied to.

Field name	Type	Required	Description
selectionId	SelectionId	✓	Selection ID of the runner in the key line handicap.
handicap	Handicap	✓	Handicap value of the key line.

KeyLineDescription

A list of KeyLineSelection objects describing the key line for the market

Field name	Type	Required	Description
keyLine	List<KeyLineSelection>	✓	A list of KeyLineSelection objects

Type Aliases

Alias	Type

MarketType	String
Venue	String
MarketId	String
SelectionId	long
Handicap	double
EventId	String
EventTypeId	String
CountryCode	String
ExchangeId	String
CompetitionId	String
Price	double
Size	double
BetId	String
MatchId	String
CustomerOrderRef	String
CustomerStrategyRef	String

Accounts API

Endpoints

Please find below the details for the current Accounts API endpoints.

If you have an **Italian** or **Spanish Exchange** account please see:

- **Betting on Italian Exchange**
- **Betting on Spanish Exchange**

Global Exchange

Interface	Endpoint	JSON-RPC Prefix	<method> Example
JSON-RPC	https://api.betfair.com/exchange/account/json-rpc/v1	<method>	AccountAPING/v1.0/getAccountFunds
JSON REST	https://api.betfair.com/exchange/account/rest/v1.0/		getAccountFunds/

Operation Summary

Required Headers

Please note - although the majority of API-NG calls require both the **X-Authentication** (sessionToken) and **X-Application** (Application Key) in the request header, this isn't applicable for some API Account Operations that are available to Software Vendors Only. The applicable headers for each Vendor API operation are included in the below table

Type	Operation	Description	Available to Software Vendors Only	X-Authentication	X-Application
DeveloperApp	createDeveloperAppKeys (String appName)	Create 2 Application Keys for given user; one 'Delayed and the other 'Live'. You must apply to have your 'Live' App Key activated.		**Required**	
List< DeveloperApp >	getDeveloperAppKeys ( )	Get all application keys owned by the given developer /vendor		**Required**	
AccountFundsResponse	getAccountFunds ( )	Get available to bet amount.		**Required**	**Required**
TransferResponse	transferFunds (Wallet from, Wallet to, double amount)			**Required**	**Required**
AccountDetailsResponse	getAccountDetails ( )	Returns the details relating your account, including your discount rate and Betfair point balance.		**Required**	**Required**
String	getVendorClientId ( )	Returns the vendor client id for customer account which is a unique identifier for that customer.		**Required**	
String	getApplicationSubscriptionToken (intsubscriptionLength)	Used to create new subscription tokens for an application. Returns the newly generated subscription token which can be provided to the end user.**Available to owner managed (Vendor) App Keys Only**	**Y**	**Required**	**Required**
Status	activateApplicationSubscription (StringsubscriptionToken)	Activates the customers subscription token for an application		**Required**	
Status	cancelApplicationSubscription (StringsubscriptionToken)	Cancel the subscription token. The customers subscription will no longer be active once cancelled**. Available to owner managed (Vendor) App Keys Only**	**Y**	**Required**	**Required**
String	updateApplicationSubscription (String vendorClientId, int subscriptionLength)	Update an application subscription with a new expiry date. **Available to owner managed (Vendor) App Keys Only**	**Y**	**Required**	**Required**
		Returns a list of subscription tokens for an application based	**Y**	**Required**	**Required**

List< ApplicationSubscription >	listApplicationSubscriptionTokens (SubscriptionStatus subscriptionStatus)	on the subscription status passed in the request.			
List< AccountSubscription >	listAccountSubscriptionTokens ( )	List of subscription tokens associated with the account. **Available to owner managed (Vendor) App Keys Only**	**Y**	**Required**	**Required**
List<SubscriptionHistory>	getApplicationSubscriptionHistory (String vendorClientId)	Returns a list of subscriptions tokens that have been associated with the customers account. **Available to owner managed (Vendor) App Keys Only**	**Y**	**Required**	**Required in request header OR request body**
AccountStatementReport	getAccountStatement **(** String locale, int fromRecord, int recordCount, TimeRange itemDateRange, IncludeItem includeItem,Walletwallet **)**	Get account statement - provides full audit trail of money moving to and from your account.	Not available via the Vendor Web API	**Required**	**Required**
List<CurrencyRate>	listCurrencyRates **(** String fromCurrency **)**	Returns a list of currency rates based on given currency.			
VendorAccessTokenInfo	**token (String client_id**, **GrantType grant_type**, String code, **String client_secret**, String refresh_token **)**	Generate web vendor session based on a standard session identifiable by auth code, vendor secret and app key	**Y**	**Required**	**Required**
VendorDetails	**getVendorDetails (String vendorId)**	Return details about a vendor from its identifier. Response includes Vendor Name and URL			
Status	**revokeAccessToWebApp (long vendorId)**	Remove the link between an account and a vendor web app. This will remove the refreshToken for this user-vendor pair subscription.			
List<VendorDetails>	**listAuthorizedWebApps ( )**	Retrieve all vendors applications currently subscribed to by the user making the request			
boolean	**isAccountSubscribedToWebApp (String vendorId)**	Return whether an account has authorised a web app.			
List<AffiliateRelation>	**getAffiliateRelation (List<String> vendorClientIds)**	Return relation between a list of users and an affiliate	**Y**	**Required**	**Required**

createDeveloperAppKeys

Operation

createDeveloperAppKeys

DeveloperApp createDeveloperAppKeys (String appName) throws AccountAPINGException

Create 2 Application Keys for given user; one 'Delayed and the other 'Live'. You must apply to have your 'Live' App Key activated.

Parameter name	Type	Required	Description
appName	String	✓	A Display name for the application.

Return type	Description
DeveloperApp	A map of application keys, one marked ACTIVE, and the other DELAYED

Throws	Description
AccountAPINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

getDeveloperAppKeys

Operation

getDeveloperAppKeys

List< DeveloperApp > getDeveloperAppKeys ( ) throws AccountAPINGException

Get all application keys owned by the given developer/vendor

Return type	Description
List< DeveloperApp >	A list of application keys owned by the given developer/vendor

Throws	Description
AccountAPINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

getAccountFunds

Operation

getAccountFunds

AccountFundsResponse getAccountFunds ( ) throws AccountAPINGException

Returns the available to bet amount, exposure and commission information.

Parameter name	Type	Required	Description
wallet	Wallet		Name of the wallet in question. Global wallet is returned by default

Return type	Description
AccountFundsResponse	Response for retrieving available to bet.

Throws	Description
AccountAPINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

transferFunds

Operation

transferFunds

TransferResponse transferFunds (Wallet from, Wallet to, double amount) throws AccountAPINGException

Transfer funds between the UK Exchange and other wallets.

This operatio is currently deprecated due to the removal of the AUS wallet

Parameter name	Type	Required	Description
from	Wallet	✓	Source wallet
to	Wallet	✓	Destination wallet
amount	double	✓	Amount to transfer

Return type	Description
TransferResponse	Response for transfer funds action

Throws	Description
AccountAPINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

getAccountDetails

Operation

getAccountDetails

AccountDetailsResponse getAccountDetails ( ) throws AccountAPINGException

Returns the details relating your account, including your discount rate and Betfair point balance.

Return type	Description
AccountDetailsResponse	Response for retrieving account details.

Throws	Description
AccountAPINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

Please note: The data returned by **getAccountDetails** relies on two underlying services. The **pointsBalance** is returned by a separate service from the other data.

As a consequence of this, in the event of a failure to a single underlying service, either the **pointsBalance** or the remaining data may not be included in the **getAccountDetails** response. If both services fail, the error UNEXPECTED_ERROR will be returned.

getAccountStatement

getAccountStatement

AccountStatementReport getAccountStatement (String locale, int fromRecord, int recordCount, TimeRange itemDateRange, includeItem, Wallet wallet**) throws AccountAPINGException**

Please see the Additional Information for details of how getAccountStatement output is affected in the event of market resettlement.

Parameter name	Type	Required	Description
locale	String		The language to be used where applicable. If not specified, the customer account default is returned.
fromRecord	int		Specifies the first record that will be returned. Records start at index zero. If not specified then it will default to 0.
recordCount	int		Specifies the maximum number of records to be returned. Note that there is a page size limit of 100.
itemDateRange	TimeRange		Return items with an itemDate within this date range. Both from and to date times are inclusive. If from is not specified then the oldest available items will be in range. If to is not specified then the latest items will be in range. nb. This itemDataRange is currently only applied when includeItem is set to ALL or not specified, else items are NOT bound by itemDate.
includeItem	IncludeItem		Which items to include, if not specified then defaults to ALL.
wallet	Wallet		Which wallet to return statementItems for. If unspecified then the UK wallet will be selected

Return type	Description
AccountStatementReport	List of statement items chronologically ordered plus moreAvailable boolean to facilitate paging

Throws	Description
AccountAPINGException	Generic exception that is thrown if this operation fails for any reason.

listCurrencyRates

listCurrencyRates

List<CurrencyRate> listCurrencyRates (String fromCurrency) throws AccountAPINGException

Returns a list of currency rates based on given currency. **Please note:** the currency rates are updated once every hour a few seconds after the hour.

Parameter name	Type	Required	Description
fromCurrency	String		The currency from which the rates are computed. **Please note:** GBP is currently the only based currency support

Return type	Description
List<CurrencyRate>	List of currency rates

Throws	Description
AccountAPINGException	Generic exception that is thrown if this operation fails for any reason.

Accounts Exceptions

Exceptions

AccountAPINGException

This exception is thrown when an operation fails

Error code	Description
INVALID_INPUT_DATA	Invalid input data
INVALID_SESSION_INFORMATION	The session token hasn't been provided, is invalid or has expired.
UNEXPECTED_ERROR	An unexpected internal error occurred that prevented successful request processing.
INVALID_APP_KEY	The application key passed is invalid or is not present
SERVICE_BUSY	The service is currently too busy to service this request
TIMEOUT_ERROR	Internal call to downstream service timed out
DUPLICATE_APP_NAME	Duplicate application name
APP_KEY_CREATION_FAILED	Creating application key version has failed
APP_CREATION_FAILED	Application creation has been failed
NO_SESSION	A session token header ('X-Authentication') has not been provided in the request
NO_APP_KEY	An application key header ('X-Application') has not been provided in the request
SUBSCRIPTION_EXPIRED	An application key is required for this operation
INVALID_SUBSCRIPTION_TOKEN	The subscription token provided doesn't exist
TOO_MANY_REQUESTS	Too many requests
INVALID_CLIENT_REF	Invalid length for the client reference
WALLET_TRANSFER_ERROR	There was a problem transferring funds between your wallets
INVALID_VENDOR_CLIENT_ID	The vendor client ID is not subscribed to this application key
USER_NOT_SUBSCRIBED	The user making the request is not subscribed to the application key they are trying to perform the action on (e.g. creating an Authorisation Code)
INVALID_SECRET	The vendor making the request has provided a vendor secret that does not match our records
INVALID_AUTH_CODE	The vendor making the request has not provided a valid authorisation cod
INVALID_GRANT_TYPE	The vendor making the request has not provided a valid grant_type, or the grant_type they have passed does not match the parameters (authCode/refreshToken)

Other parameters	Type	Required	Description
errorDetails	String		the stack trace of the error
requestUUID	String		

Accounts Enums

Enums

SubscriptionStatus

Value	Description
ALL	Any subscription status
ACTIVATED	Only activated subscriptions
UNACTIVATED	Only unactivated subscriptions
CANCELLED	Only cancelled subscriptions
EXPIRED	Only expired subscriptions

Status

Value	Description
SUCCESS	Sucess status

ItemClass

Value	Description
UNKNOWN	Statement item not mapped to a specific class. All values will be concatenated into a single key/value pair. The key will be 'unknownStatementItem' and the value will be a comma separated string. **Please note:** This is used to represent commission payment items.

Wallet

Value	Description
UK	The Global Exchange wallet

IncludeItem

Value	Description
ALL	Include all items
DEPOSITS_WITHDRAWALS	Include payments only
EXCHANGE	Include exchange bets only
POKER_ROOM	Include poker transactions only

winLose

Value	Description
RESULT_ERR	Record has been affected by a unsettlement. There is no impact on the balance for these records, this just a label to say that these are to be corrected.
RESULT_FIX	Record is a correction to the balance to reverse the impact of records shown as in error. If commission has been paid on the original settlement then there will be a second FIX record to reverse the commission.
RESULT_LOST	Loss
RESULT_NOT_APPLICABLE	Amounts relating to commission payments.
RESULT_WON	Won
COMMISSION_REVERSAL	Betfair have restored the funds to your account that it previously received from you in commission.

GrantType

Value	Description
AUTHORIZATION_CODE	Returned via the Vendor Web API token request. The **authorization code** will be valid for a single use for 10 minutes.
REFRESH_TOKEN	A token that can be used to create a new access token when using the Vendor Web API

TokenType

Value	Description
BEARER	Token type used for Vendor Web API interactions for making requests on a customers behalf.

AffiliateRelationStatus

Value	Description
INVALID_USER	Provided vendor client ID is not valid
AFFILIATED	Vendor client ID valid and affiliated
NOT_AFFILIATED	Vendor client ID valid but not affiliated

Accounts TypeDefinitions

Type definitions

TransferResponse

Transfer operation response

Field name	Type	Required	Description
transactionId	String	✓	The id of the transfer transaction that will be used in tracking the transfers between the wallets

ApplicationSubscription

Application subscription details

Field name	Type	Required	Description
subscriptionToken	String	✓	Application key identifier
expiryDateTime	Date		Subscription Expiry date
expiredDateTime	Date		Subscription Expired date
createdDateTime	Date		Subscription Create date
activationDateTime	Date		Subscription Activation date
cancellationDateTime	Date		Subscription Cancelled date
subscriptionStatus	SubscriptionStatus		Subscription status
clientReference	String		Client reference
vendorClientId	String		Vendor client Id

Subscription History

Application subscription history details

Field name	Type	Required	Description
subscriptionToken	String	✓	Application key identifier
expiryDateTime	Date		Subscription Expiry date
expiredDateTime	Date		Subscription Expired date
createdDateTime	Date		Subscription Create date
activationDateTime	Date		Subscription Activation date
cancellationDateTime	Date		Subscription Cancelled date
subscriptionStatus	SubscriptionStatus		Subscription status
clientReference	String		Client reference

AccountSubscription

Application subscription details

Field name	Type	Required	Description
subscriptionTokens	List< SubscriptionTokenInfo >	✓	Lis t of subscription token details
applicationName	String		Application name
applicationVersionId	String		Application version Id

SubscriptionTokenInfo

Subscription token information

Field name	Type	Required	Description
subscriptionToken	String	✓	Subscription token
activatedDateTime	Date		Subscription Activated date
expiryDateTime	Date		Subscription Expiry date
expiredDateTime	Date		Subscription Expired date
cancellationDateTime	Date		Subscription Cancelled date
subscriptionStatus	SubscriptionStatus		Subscription status

DeveloperApp

Describes developer/vendor specific application

Field name	Type	Required	Description
appName	String	✓	The unique name of the application
appId	long	✓	A unique id of this application
appVersions	List< DeveloperAppVersion >	✓	The application versions (including application keys)

DeveloperAppVersion

Describes a version of an external application

Field name	Type	Required	Description
owner	String	✓	The user who owns the specific version of the application
versionId	long	✓	The unique Id of the application version
version	String	✓	The version identifier string such as 1.0, 2.0. Unique for a given application.
applicationKey	String	✓	The unqiue application key associated with this application version
delayData	boolean	✓	Indicates whether the data exposed by platform services as seen by this application key is delayed or realtime.
subscriptionRequired	boolean	✓	Indicates whether the application version needs explicit subscription
ownerManaged	boolean	✓	Indicates whether the application version needs explicit management by the software owner. A value of false indicates, this is a version meant for personal developer use.
active	boolean	✓	Indicates whether the application version is currently active
vendorId	String		Public unique string provided to the Vendor that they can use to pass to the Betfair API in order to identify themselves.
vendorSecret	String		Private unique string provided to the Vendor that they pass with certain calls to confirm their identity. Linked to a particular App Key.

AccountFundsResponse

Response for retrieving available to bet.

Field name	Type	Required	Description
availableToBetBalance	double		Amount available to bet.
exposure	double		Current exposure.
retainedCommission	double		Sum of retained commission.
exposureLimit	double		Exposure limit.
discountRate	double		User Discount Rate.
pointsBalance	int		The Betfair points balance

AccountDetailsResponse

Response for Account details.

Field name	Type	Required	Description
currencyCode	String		Default user currency Code. See Currency Parameters for minimum bet sizes relating to each currency.
firstName	String		First Name.
lastName	String		Last Name.
localeCode	String		Locale Code.
region	String		Region based on users zip/postcode (ISO 3166-1 alpha-3 format). Defaults to GBR if zip/postcode cannot be identified.
timezone	String		User Time Zone.
discountRate	double		User Discount Rate.
pointsBalance	int		The Betfair points balance.
countryCode	String		The customer's country of residence (ISO 2 Char format)

AccountStatementReport

A container representing search results.

Field name	Type	Required	Description
accountStatement	List<StatementItem>	✓	The list of statement items returned by your request.
moreAvailable	boolean	✓	Indicates whether there are further result items beyond this page.

StatementItem

Summary of a cleared order.

Field name	Type	Required	Description
refId	String		An external reference, eg. equivalent to betId in the case of an exchange bet statement item.
itemDate	Date	✓	The date and time of the statement item, eg. equivalent to settledData for an exchange bet statement item. (in ISO-8601 format, not translated)
amount	double		The amount of money the balance is adjusted by
balance	double		Account balance.
itemClass	ItemClass		Class of statement item. This value will determine which set of keys will be included in itemClassData

itemClassData	Map<String, String>		Key value pairs describing the current statement item. The set of keys will be determined by the itemClass
legacyData	StatementLegacyData		Set of fields originally returned from APIv6. Provided to facilitate migration from APIv6 to API-NG, and ultimately onto itemClass and itemClassData

StatementLegacyData

Summary of a cleared order.

Field name	Type	Required	Description
avgPrice	double		The average matched price of the bet (null if no part has been matched)
betSize	double		The amount of the stake of your bet. (0 for commission payments or deposit/withdrawals)
betType	String		Back or lay
betCategoryType	String		Exchange, Market on Close SP bet, or Limit on Close SP bet.
commissionRate	String		Commission rate on market
eventId	long		**Please note:** this is the Id of the market without the associated exchangeId
eventTypeId	long		Event Type
fullMarketName	String		Full Market Name. For card payment items, this field contains the card name
grossBetAmount	double		The winning amount to which commission is applied.
marketName	String		Market Name. For card transactions, this field indicates the type of card transaction (deposit, deposit fee, or withdrawal).
marketType	marketType		Market type. For account deposits and withdrawals, marketType is set to NOT_APPLICABLE.
placedDate	Date		Date and time of bet placement
selectionId	long		Id of the selection (this will be the same for the same selection across markets)
selectionName	String		Name of the selection
startDate	Date		Date and time at the bet portion was settled
transactionType	String		Debit or credit
transactionId	long		The unique reference Id assigned to account deposit and withdrawals.
winLose	winLose		Win or loss
deadHeatPriceDivisor	double		In the instance of a dead heat, this field will indicate the number of winners involved in the dead heat (null otherwise)
avgPriceRaw	double		Currently returns same value as avgPrice. Once released will display the average matched price of the bet with no rounding applied

TimeRange

TimeRange

Field name	Type	Required	Description
from	Date		from, format: ISO 8601)
to	Date		to, format: ISO 8601

CurrencyRate

Currency rate

Field name	Type	Required	Description
currencyCode	String		Three letter ISO 4217 code
rate	double		Exchange rate for the currency specified in the request

AuthorisationResponse

AuthorisationResponse

Wrapper object containing authorisation code and redirect URL for web vendors

Field name	Type	Required	Description
authorisationCode	String	✓	The authorisation code
redirectUrl	String	✓	URL to redirect the user to the vendor page

SubscriptionOptions

SubscriptionOptions

Wrapper object containing details of how a subscription should be created

Field name	Type	Required	Description
subscription_length	int		How many days should a created subscription last for. Open ended subscription created if value not provided. Relevant only if createdSubscription is true.
subscription_token	String		An existing subscription token that the caller wishes to be activated instead of creating a new one. Ignored is createSubscription is true.
client_reference	String		Any client reference for this subscription token request.

VendorAccessTokenInfo

Wrapper object containing UserVendorSessionToken, RefreshToken and optionally a Subscription Token if one was created

Field name	Type	Required	Description
access_token	String	✓	Session token used by web vendors
token_type	TokenType	✓	Type of the token
expires_in	long	✓	How long until the token expires
refresh_token	String	✓	Token used to refresh the session token in future
application_subscription	ApplicationSubscription	✓	Object containing the vendor client id and optionally some subscription information

VendorDetails

Wrapper object containing vendor name and redirect url

Field name	Type	Required	Description
appVersionId	long	✓	Internal id of the application
vendorName	String	✓	Vendor name
redirectUrl	String		URL to be redirected to

AffiliateRelation

Wrapper object containing affiliate relation details

Field name	Type	Required	Description
vendorClientId	String	✓	ID of user
status	AffiliateRelationStatus	✓	The affiliate relation status

Navigation Data For Applications

Endpoint & Required Headers

This Navigation Data for Applications service allows the retrieval of the full Betfair market navigation menu from a compressed file.

Exchange	HTTP Method	Endpoint
UK	GET	https://api.betfair.com/exchange/betting/rest/v1/en/navigation/menu.json
ITALY	GET	https://api.betfair.it/exchange/betting/rest/v1/en/navigation/menu.json
SPAIN	GET	https://api.betfair.es/exchange/betting/rest/v1/en/navigation/menu.json

Best Practice

The file data is cached and new request for the file once an hour should be suitable for those looking to accurately recreate the Betfair navigation menu.

The following request headers are required:

- **X-Application** - Your Application Key
- **X-Authentication -** Your session token, obtained from the API login response.

Example Request

```
GET https://api.betfair.com/exchange/betting/rest/v1/en/navigation/menu.json
Connection: keep-alive
X-Application: <AppKey>
X-Authentication: <SessionToken>
Accept: application/json
Accept-Encoding: gzip,deflate
```

Supported Locales

The following languages are supported by the navigation file:

en | en_GB | bg | da | de | el | es | it | pt | ru | sv

English - en

Spanish - es

Italian - it

German - de

Swedish - sv

Portuguese -pt

Russian - ru

Greek - el

Bulgarian – bg

Danish - da

Navigation Data File Structure

This is a diagram showing how the Navigation Data File is structured.

In plain English:

A **ROOT** group node has one or many **EVENT_TYPE** nodes

An **EVENT_TYPE** node has zero, one or many **GROUP** nodes

An **EVENT_TYPE** node has zero, one or many **EVENT** nodes

A Horse Racing **EVENT_TYPE** node has zero, one or many **RACE** nodes

A **RACE** node has one or many **MARKET** nodes

A **GROUP** node has zero, one or many **EVENT** nodes

A **GROUP** node has zero, one or many **GROUP** nodes

An **EVENT** node has zero, one or many **MARKET** nodes

An **EVENT** node has zero, one or many **GROUP** nodes

An **EVENT** node has zero, one or many **EVENT** nodes

JSON Model Structure

ROOT

```
{
    "children": [
        {
            EVENT_TYPE1
        },
        {
            EVENT_TYPE2
        },
        ...
    ],
    "id": 0, // always 0
    "name": "ROOT", // always ROOT
    "type": "GROUP" // always GROUP
}
```

EVENT_TYPE

```
{
    "children": [
        {
            GROUP or EVENT or RACE (RACE only if Greyhounds/Horse Racing)
        },
        ...
    ],
    "id": "1", // Betfair specific eventTypeId
    "name": "Soccer",
    "type": "EVENT_TYPE"
}
```

GROUP

```
{
    "children": [
        {
            GROUP or EVENT
        },
        ...
    ],
    "id": "74568202414", // Not a Betfair specific id, different for every GROUP
    "name": "Womens Soccer",
    "type": "GROUP"
}
```

EVENT

```
{
    "children": [
        {
            GROUP, MARKET or EVENT
        },
        ...
    ],
    "id": "27244118", // Betfair specific eventId
    "name": "South Korea U20 (W) v Mexico U20 (W)",
    "countryCode": "GB",
    "type": "EVENT"
}
```

RACE

```
{
    "children": [
        {
            MARKET
        },
        ...
    ],
    "id": "27247020.1115", // Betfair specific raceId
    "name": "1300m 3yo",
    "startTime": "2014-08-12T11:15:00.000Z",
    "type": "RACE",
    "venue": "Deauville",
"raceNumber" : "R1", // US specific information about race numbers
    "countryCode": "GB"
}
```

MARKET

```
{
    "exchangeId": "1", // Betfair specific exchangeId
    "id": "1.114881860", // Betfair specific marketId
    "marketStartTime": "2014-08-14T00:00:00.000Z", // Betfair specific marketStartTime
    "marketType": "WIN", // Betfair specific marketType (e.g. PLACE, WIN, FORECAST etc.)
    "numberOfWinners": "2", // Betfair specific number of winners
    "name": "Over/Under 6.5 Goals",
    "type": "MARKET"
}
```

Heartbeat API

This Heartbeat operation is provided to allow customers to automatically cancel their unmatched bets in the event of their API client/s losing connectivity with the Betfair API.

UK Exchange

Interface	Endpoint	<method> Example
JSON-RPC	https://api.betfair.com/exchange/heartbeat/json-rpc/v1	HeartbeatAPING/v1.0/heartbeat

Italian Exchange

Interface	Endpoint	<method> Example
JSON-RPC	https://api.betfair.it/exchange/heartbeat/json-rpc/v1	HeartbeatAPING/v1.0/heartbeat

Spanish Exchange

Interface	Endpoint	<method> Example
JSON-RPC	https://api.betfair.es/exchange/heartbeat/json-rpc/v1	HeartbeatAPING/v1.0/heartbeat

Operation summary

HeartbeatReport	**heartbeat (int preferredTimeoutSeconds)**

Detailed documentation

Heartbeat

Operations

heartbeat

HeartbeatReport heartbeat (int preferredTimeoutSeconds) throws APINGException

This heartbeat operation is provided to help customers have their positions managed automatically in the event of their API clients losing connectivity with the Betfair API. If a heartbeat request is not received within a prescribed time period, then Betfair will attempt to cancel all 'LIMIT' type bets for the given customer on the given exchange. There is no guarantee that this service will result in all bets being cancelled as there are a number of circumstances where bets are unable to be cancelled. Manual intervention is strongly advised in the event of a loss of connectivity to ensure that positions are correctly managed. If this service becomes unavailable for any reason, then your heartbeat will be unregistered automatically to avoid bets being inadvertently cancelled upon resumption of service. you should manage your position manually until the service is resumed. Heartbeat data may also be lost in the unlikely event of nodes failing within the cluster, which may result in your position not being managed until a subsequent heartbeat request is received.

Parameter name	Type	Required	Description
preferredTimeoutSeconds	int	✓	Maximum period in seconds that may elapse (without a subsequent heartbeat request), before a cancellation request is automatically submitted on your behalf. **The minimum value is 10, the maximum value permitted is 300**. Passing 0 will result in your heartbeat being unregistered (or ignored if you have no current heartbeat registered). You will still get an actionPerformed value returned when passing 0, so this may be used to determine if any action was performed since your last heartbeat, without actually registering a new heartbeat. Passing a negative value will result in an error being returned, INVALID_INPUT_DATA. Any errors while registering your heartbeat will result in a error being returned, UNEXPECTED_ERROR. Passing a value that is less than the minimum timeout will result in your heartbeat adopting the minimum timeout. Passing a value that is greater than the maximum timeout will result in your heartbeat adopting the maximum timeout. The minimum and maximum timeouts are subject to change, so your client should utilise the returned actualTimeoutSeconds to set an appropriate frequency for your subsequent heartbeat requests.

Return type	Description
HeartbeatReport	Response from heartbeat operation

Throws	Description
APINGException	Thrown if the operation fails

Events

This interface does not define any events.

Type definitions

HeartbeatReport

Response from heartbeat operation

Field name	Type	Required	Description
actionPerformed	ActionPerformed	✓	The action performed since your last heartbeat request.
actualTimeoutSeconds	int	✓	The actual timeout applied to your heartbeat request, see timeout request parameter description for details.

Enums

ActionPerformed

Value	Description
NONE	No action was performed since last heartbeat, or this is the first heartbeat
CANCELLATION_REQUEST_SUBMITTED	A request to cancel all unmatched bets was submitted since last heartbeat
ALL_BETS_CANCELLED	All unmatched bets were cancelled since last heartbeat
SOME_BETS_NOT_CANCELLED	Not all unmatched bets were cancelled since last heartbeat
CANCELLATION_REQUEST_ERROR	There was an error requesting cancellation, no bets have been cancelled
CANCELLATION_STATUS_UNKNOWN	There was no response from requesting cancellation, cancellation status unknown

Exceptions

APINGException

This exception is thrown when an operation fails

Error code	Description
INVALID_INPUT_DATA	Invalid input data
INVALID_SESSION_INFORMATION	The session token passed is invalid
NO_APP_KEY	An application key is required for this operation
NO_SESSION	A session token is required for this operation
INVALID_APP_KEY	The application key passed is invalid
UNEXPECTED_ERROR	An unexpected internal error occurred that prevented successful request processing.

Other parameters	Type	Required	Description
errorDetails	String		Specific error details
requestUUID	String		

Typical Interaction

SET UP

```
[{"jsonrpc": "2.0", "method": "HeartbeatAPING/v1.0/heartbeat", "params": {"preferredTimeoutSeconds":"10"}, "id": 1}]
```

```
 [{"jsonrpc":"2.0","result":{"actualTimeoutSeconds":10,"actionPerformed":"NONE"},"id":1}]
```

RESET

```
 [{"jsonrpc": "2.0", "method": "HeartbeatAPING/v1.0/heartbeat", "params": {"preferredTimeoutSeconds":"0"}, "id": 1}]
```

```
 [{"jsonrpc":"2.0","result":{"actualTimeoutSeconds":0,"actionPerformed":"NONE"},"id":1}]
```

RE-SET UP HEARTBEAT

```
 [{"jsonrpc": "2.0", "method": "HeartbeatAPING/v1.0/heartbeat", "params": {"preferredTimeoutSeconds":"10"}, "id": 1}]
```

```
 [{"jsonrpc":"2.0","result":{"actualTimeoutSeconds":10,"actionPerformed":"NONE"},"id":1}]
```

You should be able to reset the heartbeat by passing a value of actualTimeoutSeconds":0 and then restarting it by setting the required value.

EXAMPLE OF RESPONSE IF HEARTBEAT ISN'T RECEIVED WITHIN SPECIFIED TIME

```
[{"jsonrpc": "2.0", "method": "HeartbeatAPING/v1.0/heartbeat", "params": {"preferredTimeoutSeconds":"10"}, "id": 1}]
```

```
[{"jsonrpc":"2.0","result":{"actualTimeoutSeconds":10,"actionPerformed":"ALL_BETS_CANCELLED"},"id":1}]
```

Race Status API

Fhar

The **listRaceDetails** operation is provided to allow customers to establish the status of a horse or greyhound race market both prior to and after the start of the race. This information is available for UK and Ireland races only and is provided for information purposes only. **We cannot provide any guarantees regarding the timeliness of the data vs live /broadcasted pictures.**

listRaceDetails

Interface	Endpoint	JSON-RPC Prefix	<method> Example
JSON-RPC	https://api.betfair.com/exchange/scores/json-rpc/v1	<method>	ScoresAPING/v1.0/listRaceDetails

Operation summary

List<RaceDetails>	**listRaceDetails (** Set<MeetingId> meetingIds, Set<RaceId> raceIds **)**

Operations

listRaceDetails

List<RaceDetails> listRaceDetails (Set<MeetingId> meetingIds, Set<RaceId> raceIds **) throws APINGException**

Search for races to get their details.

Parameter name	Type	Required	Description

meetingIds	Set<MeetingId>		Optionally restricts the results to the specified meeting IDs. The unique Id for the meeting equivalent to the eventId for that specific race as returned by listEvents. Optionally restricts the results to the specified meeting IDs.
raceIds	Set<RaceId>		Optionally restricts the results to the specified race IDs. The unique Id for the race in the format meetingid.raceTime (hhmm). raceTime is in GMT. Optionally restricts the results to the specified race IDs. The raceID field is returned within the Race node of the Navigation Data For Applications service.

Return type	Description
List<RaceDetails>	List of retrieved race details

Throws	Description
APINGException	

Since 1.0.0

Events

This interface does not define any events.

Type definitions

RaceDetails

Race Details

Field name	Type	Required	Description
meetingId	MeetingId	✓	The unique Id for the meeting equivalent to the eventId for that specific race as returned by listEvents. Optionally restricts the results to the specified meeting IDs.
raceId	RaceId	✓	The unique Id for the race in the format meetingid.raceTime (hhmm). Optionally restricts the results to the specified race IDs.
raceStatus	RaceStatus	✓	The current status of the race.
lastUpdated	LastUpdated		This is the time the data was last updated

responseCode	ResponseCode	✓	

Type aliases

Alias	Type
UpdateSequence	long
EventId	String
EventTypeId	String
EventTime	String
UpdateType	String
LastUpdated	Date
MeetingId	String
RaceId	String

Enums

`RaceStatus`

Value	Description
DORMANT	There is no data available for this race.
DELAYED	The start of the race has been delayed
PARADING	The horses/greyhounds are in the parade ring
GOINGDOWN	The horses are going down to the starting post
GOINGBEHIND	The horses are going behind the stalls
APPROACHING	The greyhounds are approaching the traps
GOINGINTRAPS	The greyhounds are being put in the traps
HARERUNNING	The hare has been started
ATTHEPOST	The horses are at the post

OFF	The greyhound/horse race has started
FINISHED	The race has finished
FINALRESULT	The result has been declared (Greyhounds only)
FALSESTART	There has been a false start
PHOTOGRAPH	The result of the race is subject to a photo finish
RESULT	The result of the race has been announced
WEIGHEDIN	The jockeys have weighed in
RACEVOID	The race has been declared void
NORACE	The race has been declared a no race
MEETINGABANDONED	The meeting has been abandoned
RERUN	The race will be rerun
ABANDONED	The race has been abandoned

ResponseCode

Value	Description
OK	Data returned successfully
NO_NEW_UPDATES	No updates since the passes UpdateSequence
NO_LIVE_DATA_AVAILABLE	Event scores are no longer available or are not on the schedule
SERVICE_UNAVAILABLE	Data feed for the event type (tennis/football etc) is currently unavailable
UNEXPECTED_ERROR	An unexpected error occurred retrieving score data

LIVE_DATA_TEMPORARILY_UNAVAILABLE	Live Data feed for this event/match is temporarily unavailable, data could potentially be stale

Exceptions

APINGException

This exception is thrown when an operation fails

Error code	Description
UNEXPECTED_ERROR	The operation failed with an unexpected error
INVALID_INPUT_DATA	Invalid input data
INVALID_SESSION_INFORMATION	The session token passed is invalid or expired
INVALID_APP_KEY	The application key passed is invalid
SERVICE_BUSY	The service is currently too busy to service this request
TIMEOUT_ERROR	Internal call to downstream service timed out
NO_SESSION	A session token is required for this operation
NO_APP_KEY	An application key is required for this operation
TOO_MANY_REQUESTS	Too many requests
SERVICE_UNAVAILABLE	Service is currently unavailable

Other parameters	Type	Required	Description
errorDetails	String		The stack trace of the error
requestUUID	String		

Vendor Services API

When & How to Use Vendor Services

There are two separate methods available to Vendors.

- **Vendor Services API** - Available to licensed software vendors who want to provide Betfair Exchange customers access to their certified software using a subscription based system.
- **Vendor Web API** - Available to licensed software vendors who want to develop **web applications** using the Betfair API for access by Betfair Exchange customers.

Vendor Services API

The Vendor Services API allows licensed and certified Software Vendors to manage the permissioning of Betfair accounts to their API-NG application/s.

With the Vendor Services API, you can:

- Grant permission for users to access your application
- Set the permission to expire after a specific date
- Store arbitrary information against each subscription
- View existing and historical application subscriptions

- For details of the operations to manage subscriptions, please see the Accounts section of the API-NG Reference Guide.

There are two key processes that Vendor will need to following when using the Vendor Services in API-NG:

- Subscribing a Customer.
- Checking a Customers Subscription History.

Each of these processes are detailed below.

Subscribing a Customer

Preconditions:

- Customer has a Betfair account.
- Vendor has a Betfair vendor account, the App Key for their app, and their (Vendor) session token

Example flow:

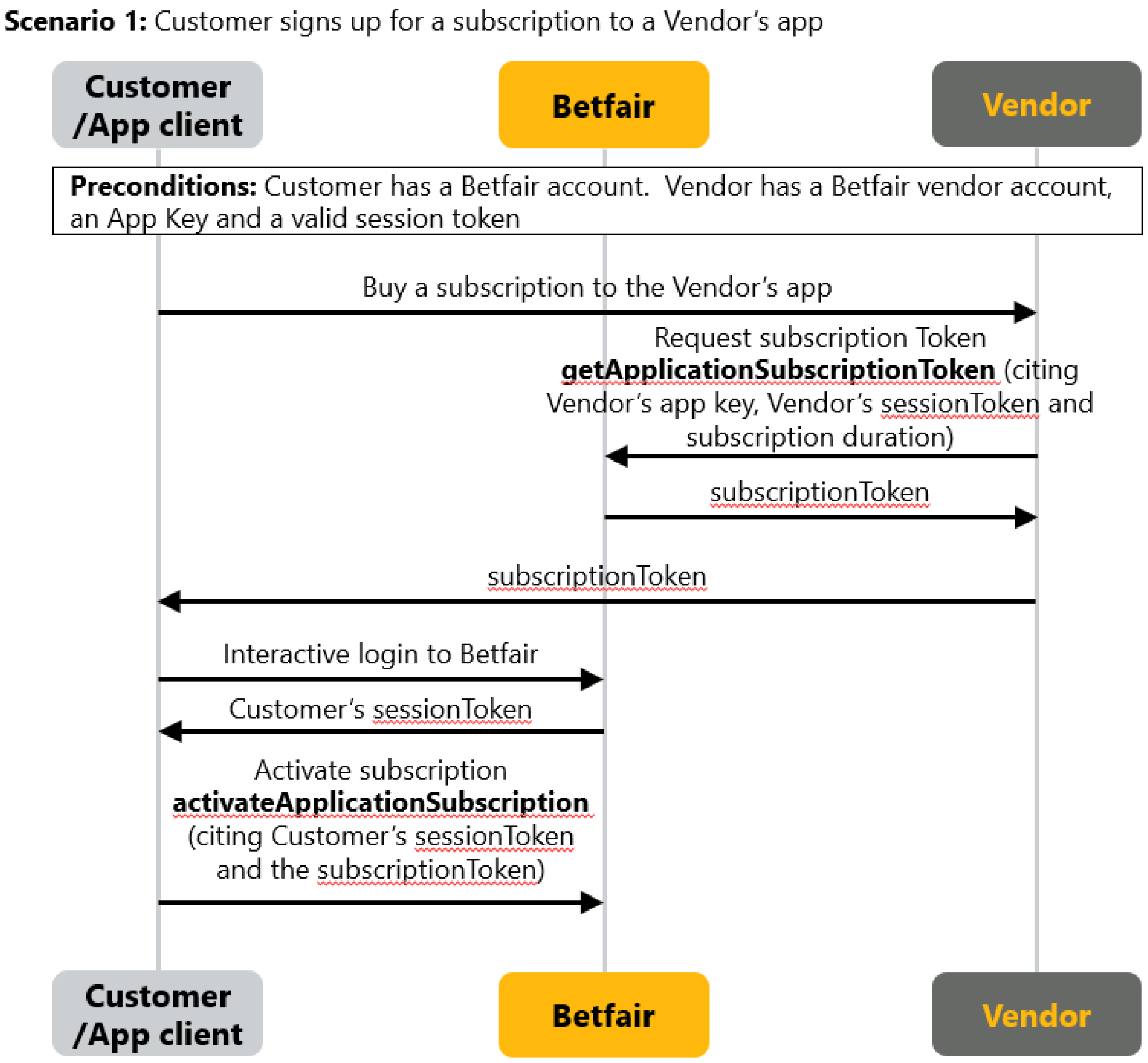
1. Customer visits Vendor’s web site and decides to sign up for the Vendor’s app.
2. The Customer purchases a 12 month subscription to the app from the Vendor’s web site.
3. The Vendor’s website server then calls the Account API-NG operation **getApplicationSubscriptionToken**, passing in the Vendor’s App Key, a valid Session Token for the Vendor’s Betfair account (to prove that they own that App Key) and the length of the subscription required (**365 days in this scenario**).
4. Betfair returns a Subscription Token in the form of **ABCD-EFGH-IJKL** to the Vendor
5. The Subscription Token is then provided to the instance of the Vendor’s app used by the Customer.

   This may happen in a number of ways, including:

   i) Emailed from Vendor to Customer, who then types the Subscription Token into the app.

ii) Associated with the Customer's app instance by the Vendor, so that the token can be cited by the app based on Customer login to the app.

6. The Vendors app requires the Customer to login to Betfair using the Interactive Login method. See sample code Interactive Login sample code her
7. Betfair returns Customer's Session Token to the app.
8. The subscription is then activated by the app calling **activateApplicationSubscription** passing in the **Subscription Token** and a valid **Session Token** for the Customer. **Please note:** the X-Application header is not required for the **activateApplicationSubscription** request.


Scenario 1: Customer signs up for a subscription to a Vendor's app
Customer /App client
Betfair
Vendor
Preconditions: Customer has a Betfair account. Vendor has a Betfair vendor account, an App Key and a valid session token
Buy a subscription to the Vendor's app
Request subscription Token **getApplicationSubscriptionToken** (citing Vendor's app key, Vendor's sessionToken and subscription duration)
subscriptionToken
subscriptionToken
Interactive login to Betfair
Customer's sessionToken
Activate subscription **activateApplicationSubscription** (citing Customer's sessionToken and the subscriptionToken)
Customer /App client
Betfair
Vendor

Checking a Customers Subscription History

In this scenario the Vendor wishes to check the customer's subscription status to establish if the customer has:

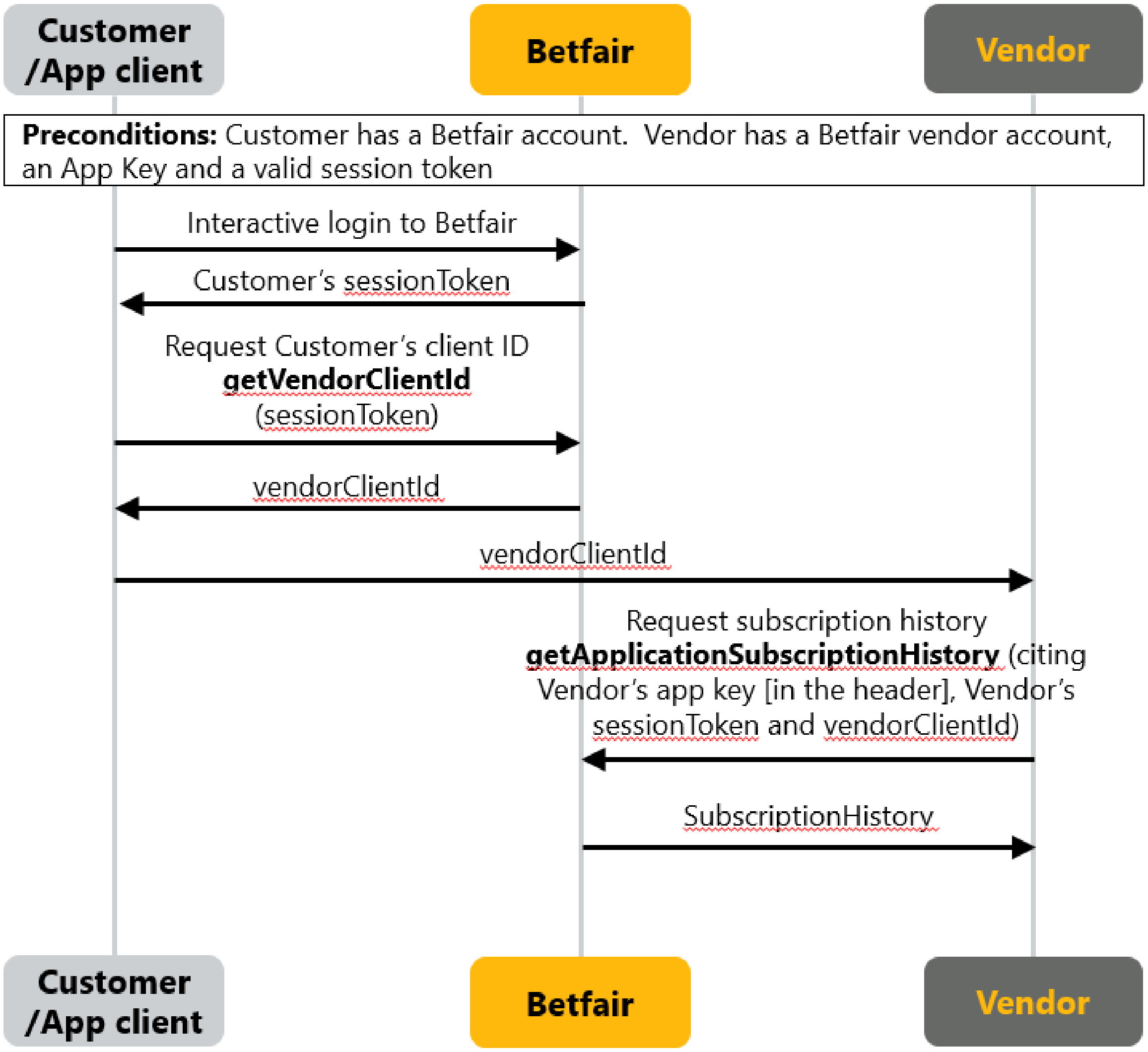
1. An existing active subscription.
2. An expired subscription.
3. A cancelled subscription.
4. No current or previous subscription history (i.e. Customer is entirely new).

This can be done within the application itself (client side) or by Vendor on their app server (server side). Each process is explained in more detail below:

Client side process:

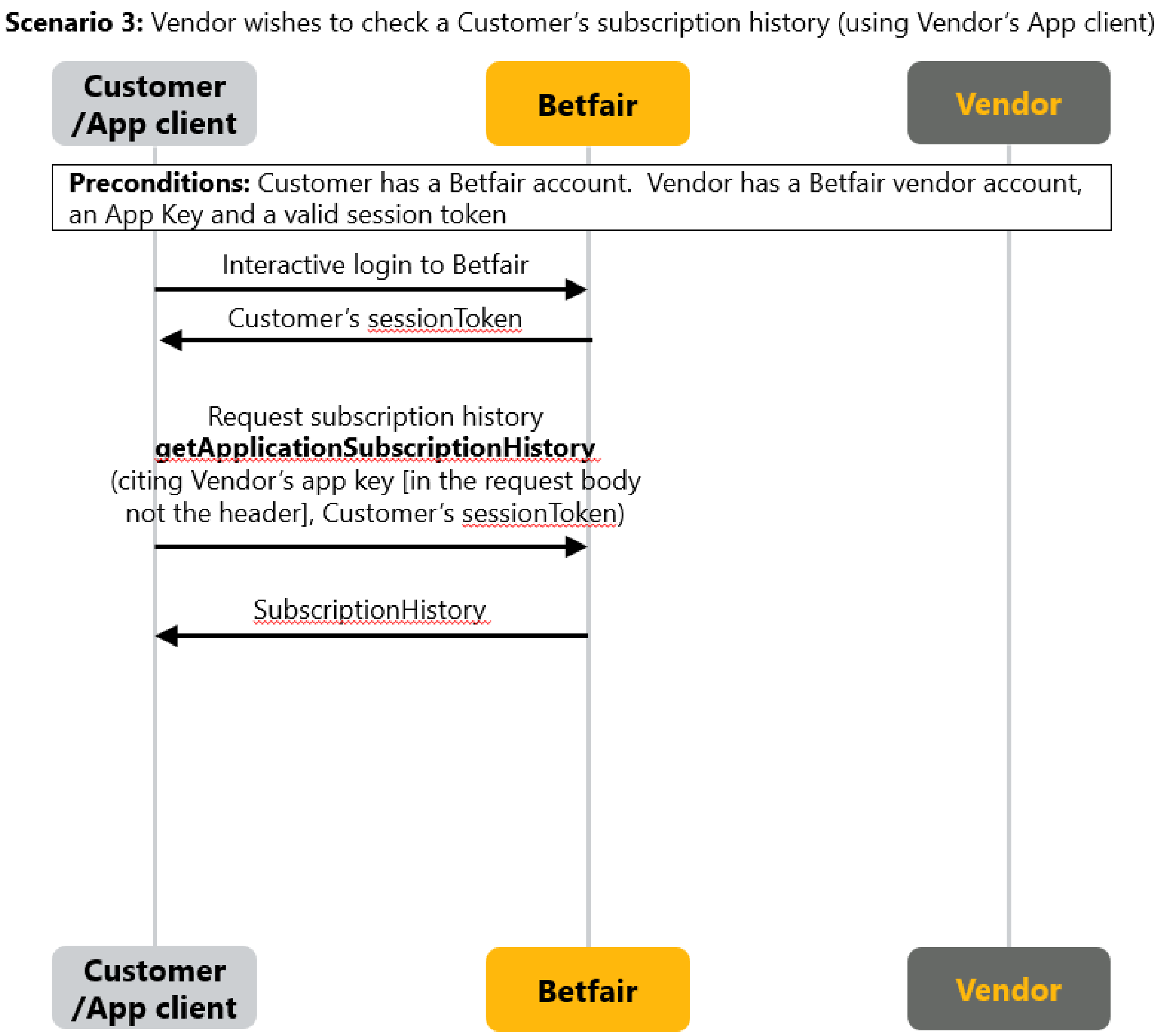
1. Vendor's app client requires Customer to login to Betfair using the Interactive Login method.
2. Betfair returns Customer's Session Token to the Vendor's app client.
3. Vendor's app client calls **getApplicationSubscriptionHistory** citing Vendors App Key in the request body and the customer's Session Token in the X-Authentication header.
4. Betfair returns Customer's subscription history. If an empty list is returned then the customer has no current or previous subscription history (i.e. Customer is entirely new)

Scenario 2: Vendor wishes to check a Customer's subscription history (using Vendor's App Server)


Customer /App client
Betfair
Vendor
Preconditions: Customer has a Betfair account. Vendor has a Betfair vendor account, an App Key and a valid session token
Interactive login to Betfair
Customer's sessionToken
Request Customer's client ID getVendorClientId (sessionToken)
vendorClientId
vendorClientId
Request subscription history getApplicationSubscriptionHistory (citing Vendor's app key [in the header], Vendor's sessionToken and vendorClientId)
SubscriptionHistory
Customer /App client
Betfair
Vendor

Server side process:

1. Vendor's app client requires Customer to login to Betfair using Interactive Login method.
2. Betfair returns Customer's Session Token to the Vendor's app client.
3. Vendor's app client calls **getVendorClientId** citing Customer's session token from step 1.
4. Betfair returns Customer's vendorClientId.
5. Vendor's App client sends Customer's vendorClientId to Vendor's app server.
6. Vendor's app server calls **getApplicationSubscriptionHistory** citing Vendor's Session Token, Vendor's App Key, and the vendorClientID from step 4.
7. Betfair returns Customer's complete history for the Vendor's app (as identified by the app key cited in step 5). If an empty list is returned then the customer has no current or previous subscription history (i.e. Customer is entirely new).


Scenario 3: Vendor wishes to check a Customer's subscription history (using Vendor's App client)
Customer /App client
Betfair
Vendor
Preconditions: Customer has a Betfair account. Vendor has a Betfair vendor account, an App Key and a valid session token
Interactive login to Betfair
Customer's sessionToken
Request subscription history getApplicationSubscriptionHistory (citing Vendor's app key [in the request body not the header], Customer's sessionToken)
SubscriptionHistory
Customer /App client
Betfair
Vendor

Vendor Web API

The Vendor Web API is available to licensed Software Vendors who are creating **web based applications** These operations enable the web application to carry out operations on the users behalf using the OAuth2 protocol.

Operations

The following operations are available via the Vendor Web API

- token
- revokeAccessToWebApp
- isAccountSubscribedToWebApp

OAuth 2 Flow - Creating an Access Token

The user authentication via web apps is underpinned by the OAuth 2 protocol.

You will need to redirect the user to the Betfair login page (identitysso.betfair.com/view/vendor-login), along with up to three parameters:

- **client_id (mandatory):** this is your vendor ID, returned to you by the **getDeveloperAppKeys** call provided you have been registered as a web vendor
- **response_type**: this needs to be set to **'code'**, to indicate you are requesting an authorization code (see later)
- **redirect_uri (optional):** this will be appended to the URL you provided **upon sign up to the Vendor Program** when we redirect the user back to your site after login. The **redirect_uri** needs to be URL encoded.

Example call - Creating An Access Token

```
https://identitysso.betfair.com/view/vendor-login?client_id=123456&response_type=code&redirect_uri=newjoiner
```

Linking OAuth 2 Flow to Affiliate PartnerId

If you're a **Betfair Affiliate**, you should append the following to the **redirect_uri** to ensure that any new account sign ups are linked to your assigned partnerId '**&rfr=xxxx&PI=xxxx&pi=partnerxxxx'**

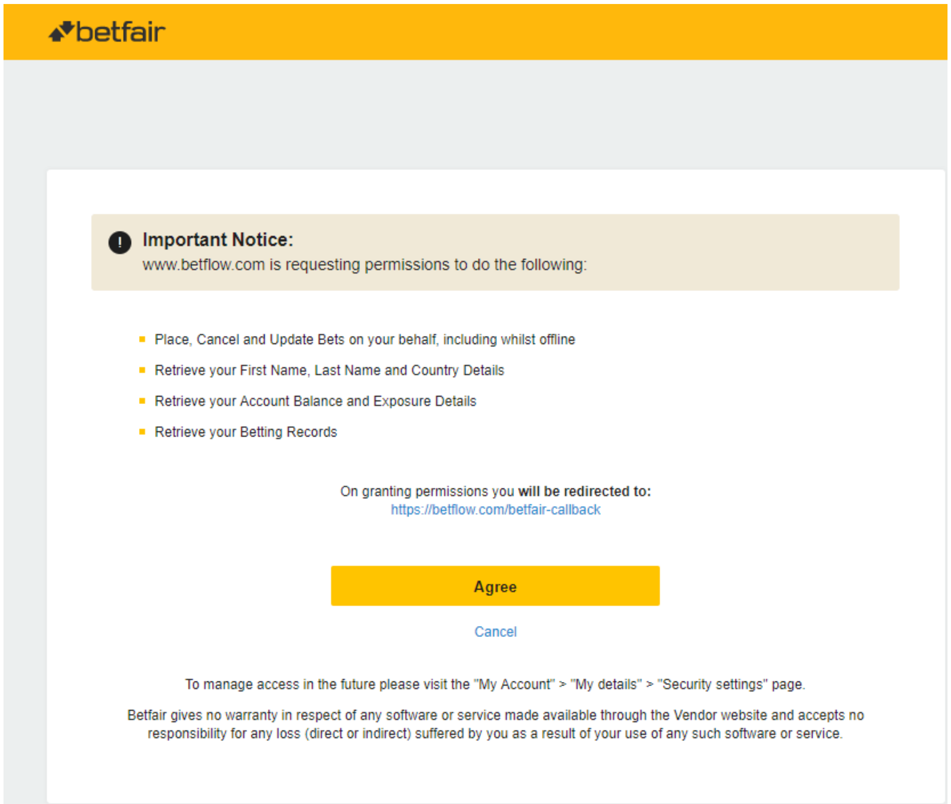
Please replace 'xxxx' with your own partnerId

For example: https://identitysso.betfair.com/view/vendor-login?client_id=123456&response_type=code&redirect_uri=newjoiner**&rfr=12345&PI=12345&pi=partner12345**

Then Encode the redirect_uri value: https://identitysso.betfair.com/view/vendor-login?client_id=123456&response_type=code&redirect_uri=**newjoiner%26rfr%3D12345%26PI%3D12345%26pi%3Dpartner12345**

This ensures that your partner id persists through the Betfair registration should a new user need to open a Betfair account via Join Now

The user will log in and be presented with a message asking them to allow you to perform Betfair calls on their behalf, as shown below.



Important Notice:
www.betflow.com is requesting permissions to do the following:

- Place, Cancel and Update Bets on your behalf, including whilst offline
- Retrieve your First Name, Last Name and Country Details
- Retrieve your Account Balance and Exposure Details
- Retrieve your Betting Records

On granting permissions you **will be redirected to:**
https://betflow.com/betfair-callback

To manage access in the future please visit the "My Account" > "My details" > "Security settings" page.

Betfair gives no warranty in respect of any software or service made available through the Vendor website and accepts no responsibility for any loss (direct or indirect) suffered by you as a result of your use of any such software or service.

Should they accept this message, they will be redirected to your site (using your **redirect URL + any optional parameters**). Critically, we will include an **authorization code** as a parameter, under the parameter name "code".

NOTE: The **authorization code** will be valid for a single use for 10 minutes.

Example redirect

```
https://mywebsite.com/newjoiner?code=12345
```

You will need to propagate this **'code'** to your back-end, from which you will have to exchange it for an access token.

The **access token** will allow you to use the Betfair API on the user's behalf. To obtain the **access token** you will need to call the token operation on the Accounts API

Example Token Request

The token call takes the following parameters:

Request Headers

- **Header**: 'X-Application' : 'Your app key' **(This is the App Key assigned to your Vendor account)**
- **Header**: 'X-Authentication' : 'Your session token' **(This the session token generated by your Vendor account)**

Request Body

- **client_id:** your vendor ID, returned to you by the **getDeveloperAppKeys** call provided you have been registered as a web vendor
- **grant_type**: in this context, this will have to be set to 'AUTHORIZATION_CODE'
- **code**: the code you were returned
- **client_secret**: your secret, obtained from GetDeveloperAppKeys

Response Parameters

- **access_token**: the access token, used to call Betfair on the user's behalf
- **token_type**: meta data for the access token (see 'Making calls on the user's behalf')
- **expires_in**: how long the access token will be valid for (in seconds)
- **refresh_token**: a token that can be used to create a new access token (see 'Using the refresh token')
- **application_subscription**: contains the vendor client ID, a unique identifier for a user. Can also contain some subscription related information (See 'Legacy Subscriptions')

IMPORTANT

To protect sensitive information such as your app key and secret, it is important that the token operation only be called from server to server.

Example token request - REST

```
X-Application: 'your App key'
X-Authentication : 'your session token'
Accept:  application/json
Content-Type:  application/json

Endpoint

https://api.betfair.com/exchange/account/rest/v1.0/token/

Request Body

{"client_id"="4534","grant_type":"AUTHORIZATION_CODE","code":"-22a1-12151008-000007cb61","client_secret":"
bc183d-f5-40dc-82a6-d97681"}

Response

{"access_token":"KeOi+kyg2RvDK4HM+W46CvSnP5w=","refresh_token":"50d76117-7f85-375v-a38f-ffb332713f93","
application_subscription":{"vendor_client_id":"456238"},"token_type":"BEARER","expires_in":"14400"}
```

Example token request - JSON

```
X-Application: 'your App key'
X-Authentication : 'your session token'
Accept:  application/json
Content-Type:  application/json

Endpoint

https://api.betfair.com/exchange/account/json-rpc/v1

Request

{"jsonrpc": "2.0", "method": "AccountAPING/v1.0/token", "params": {"client_id":"CLIENTID","grant_type":"
AUTHORIZATION_CODE","code":"CODE","client_secret":"CLIENTSECRET"}, "id": 1 }
```

OAuth 2 Flow - Using the Refresh Token

When the access token expires, it is possible to create a new one without any user input, using the refresh token. This is done with the same call that was used to create it originally, the token operation, but with a different set of parameters, for example, grant_type: REFRESH_TOKEN and refresh_token:

Header: 'X-Application' : 'your app key'

Header: 'X-Authentication' : 'your session token'

client_id: your vendor ID, obtained from GetDeveloperAppKeys

grant_type: 'REFRESH_TOKEN'

refresh_token: the refresh token for the **user**

client_secret: your secret, obtained from GetDeveloperAppKeys

This will return the same information as the original call:

access_token: the access token, used to call Betfair on the user's behalf

token_type: meta data for the access token (see 'Making calls on the user's behalf')

expires_in: how long the access token will be valid for (in seconds)

refresh_token: the refresh token remains the same

application_subscription: contains the vendor client ID, a unique identifier for a user

Example Refresh Token request - REST

```
X-Application: 'your App key'
X-Authentication : 'your session token'
Accept:  application/json
Content-Type:  application/json

Endpoint

https://api.betfair.com/exchange/account/rest/v1.0/token/

Request Body

{"client_id"="4534","grant_type":"REFRESH_TOKEN","refresh_token":"50d76117-7f85-375v-a38f-ffb332713f93","
client_secret":"bc183d-f5-40dc-82a6-d97681"}

Response

{"access_token":"KeOi+kyg2RvDK4HM+W46CvSnP5w=","refresh_token":"50d76117-7f85-375v-a38f-ffb332713f93","
application_subscription":{"vendor_client_id":"456238"},"token_type":"BEARER","expires_in":"14400"}
```

You can user the 'expires_in' value to determine when the access token will stop being valid. Alternatively, if calls made with the access token start returning an INVALID_SESSION error, it is likely that the token has expired.

Making API Calls On The Users Behalf

Once you have an access token, you can start calling the Betfair API on the user's behalf based on their actions on your site.

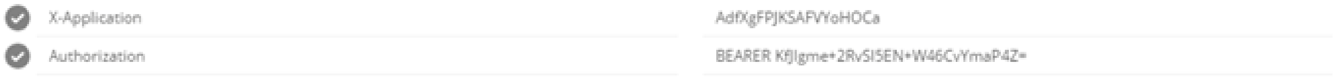
You will need to populate headers in a different way to a standard API call:

X-Application: your application key

Authorization: token_type + " " + access_token

The Authorization header needs to be a concatenation of the token type and the access token (both returned by the token call), separated by a space.

Example:



✓	X-Application	AdfXgFPJKSAFVYoHOCa
✓	Authorization	BEARER KfJlgme+2RvSI5EN+W46CvYmaP4Z=

Betfair will use the access token to determine which user the calls are being made for.

These calls also have to be from back-end to back-end.

Please Note

The following API operations aren't available using the Vendor Web API

- Navigation Data for Applications.
- getAccountStatement

User Revocation

The user may choose to revoke the permissions previously granted to your web application. **You must provide the user with the facility to do so within any web app using the revokeAccessToWebApp operation.**

This will invalidate the access token and destroy your refresh token. Any subsequent calls to Betfair using the access token, or any attempt to generate a new one using the refresh token will result in either an **INVALID_SESSION** exception or **UNEXPECTED_ERROR** respectively.

Legacy Subscriptions

The way subscriptions are handled for web applications differ greatly from the way they were for desktop based applications. The subscription token model is no longer enforced, however you may still choose to create and manage subscriptions using the existing API calls.

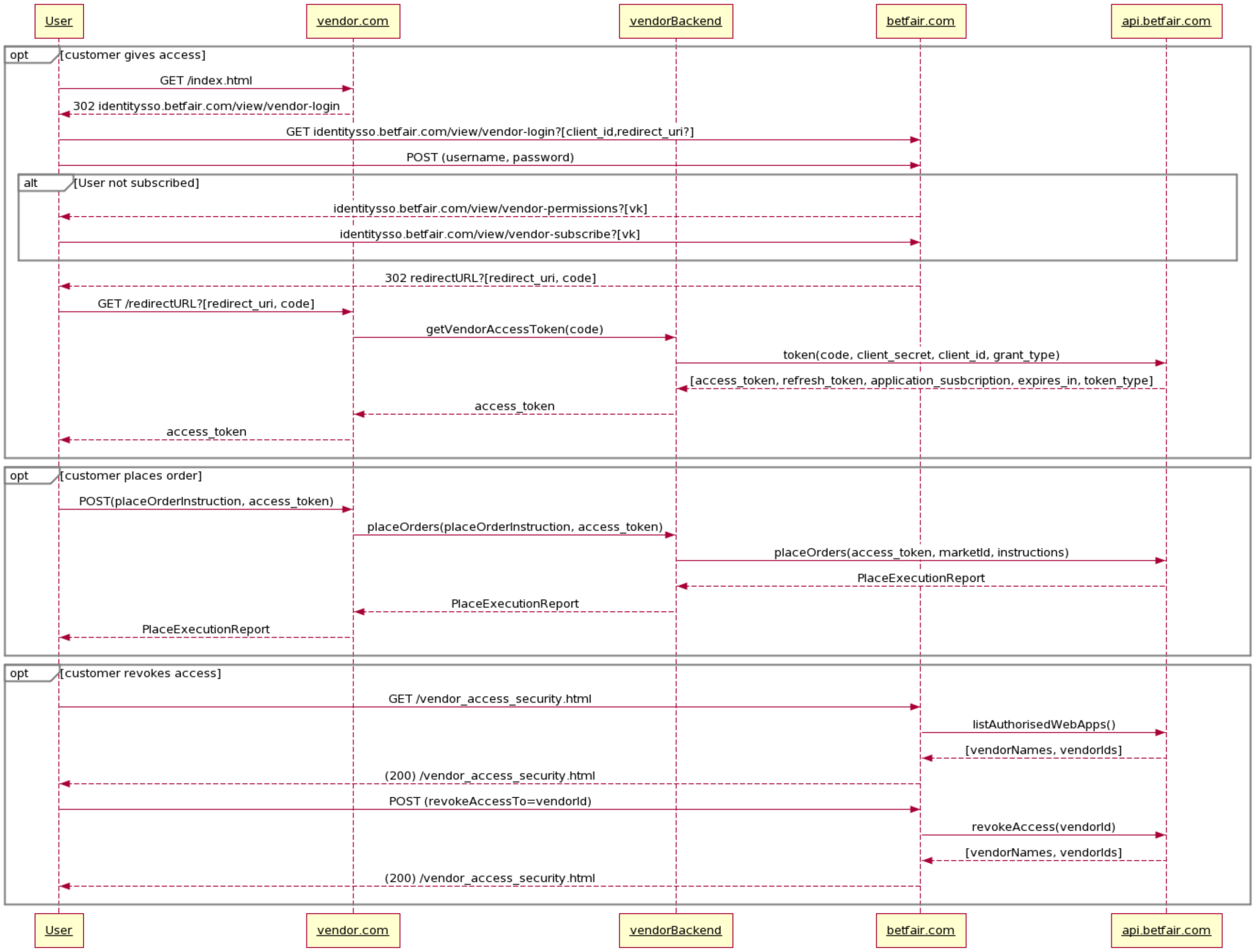
Please note that in this model, Betfair will no-longer prevent requests for a user with an expired or cancelled subscription. You can choose what action to take based on information contained in the application_subscription field returned by the token operation.

The available operations are the following:

- getApplicationSubscriptionToken
- activateApplicationSubscription
- cancelApplicationSubscription
- updateApplicationSubscription
- listApplicationSubscriptionTokens
- getApplicationSubscriptionHistory

If you do use these calls to manage subscriptions, the token call will return information on the most relevant subscription (i.e. the subscription with the latest expiration date) as part of the application_subscription object.

Flow Diagram


User
vendor.com
vendorBackend
betfair.com
api.betfair.com
opt
[customer gives access]
GET /index.html
302 identitysso.betfair.com/view/vendor-login
GET identitysso.betfair.com/view/vendor-login?[client_id,redirect_uri?]
POST (username, password)
alt
[User not subscribed]
identitysso.betfair.com/view/vendor-permissions?[vk]
identitysso.betfair.com/view/vendor-subscribe?[vk]
302 redirectURL?[redirect_uri, code]
GET /redirectURL?[redirect_uri, code]
getVendorAccessToken(code)
token(code, client_secret, client_id, grant_type)
[access_token, refresh_token, application_susbcription, expires_in, token_type]
access_token
access_token
opt
[customer places order]
POST(placeOrderInstruction, access_token)
placeOrders(placeOrderInstruction, access_token)
placeOrders(access_token, marketId, instructions)
PlaceExecutionReport
PlaceExecutionReport
PlaceExecutionReport
opt
[customer revokes access]
GET /vendor_access_security.html
listAuthorisedWebApps()
[vendorNames, vendorIds]
(200) /vendor_access_security.html
POST (revokeAccessTo=vendorId)
revokeAccess(vendorId)
[vendorNames, vendorIds]
(200) /vendor_access_security.html
User
vendor.com
vendorBackend
betfair.com
api.betfair.com

Sample Code

Sample code that demonstrates the web apps interaction is available via https://github.com/betfair/sample-web-app-vendor/

getVendorClientId

getVendorClientId

String getVendorClientId ( ) throws AccountAPINGException

Returns the vendor client id for customer account which is a unique identifier for that customer. The vendor client Id can be used to obtain the customers application subscription history via getApplicationSubscriptionHistory. The request requires the X-Authentication header only

Return type	Description
String	Vendor client id.

Throws	Description
AccountAPINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

activateApplicationSubscription

Operation

activateApplicationSubscription

Status activateApplicationSubscription (String subscriptionToken) throws AccountAPINGException

Activates the customers subscription token for an application. **Please note:** The request is made by the customers account using their session token (X-Authentication header) only.

- The activation of a new subscription token can take up to 2 minutes, therefore, you should ensure that this delay is handled within your application.
- **Please note:** A maximum number of **15,000** subscription **UNACTIVATED** subscriptions tokens can be created at any one time. Attempts to create more subscription tokens will return the error **TOO_MANY_REQUESTS** error which will restrict creation of further tokens until existing **UNACTIVATED** subscription tokens have been **ACTIVATED** or **CANCELLED.**

Parameter name	Type	Required	Description
subscriptionToken	String	✓	Subscription token for activation.

Return type	Description
Status	

Throws	Description
AccountAPINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

cancelApplicationSubscription

Operation

cancelApplicationSubscription

Status cancelApplicationSubscription (String subscriptionToken) throws AccountAPINGException

Cancel the subscription token. The customers subscription will no longer be active once cancelled.

Parameter name	Type	Required	Description
subscriptionToken	String	✓	Subscription token to cancel

Return type	Description
Status	

Throws	Description
AccountAPINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

updateApplicationSubscription

Operation

updateApplicationSubscription

String updateApplicationSubscription (String vendorClientId, **int subscriptionLength) throws AccountAPINGException**

Update an application subscription with a new expiry date. **Please note**: A new subscription token will be created and existing tokens will be cancelled automatically

Please note: A subscription token created by this operation ***doesn't*** need to be activated via **activateApplicationSubscription** as the token is automatically associated with the customers vendorClientId when the request is made.

Parameter name	Type	Required	Description
vendorClientId	String	✓	The vendor client id for which to update the subscription for
subscriptionLength	int	✓	How many days the subscription should last. Expiry time will be rounded up to midnight on the date of expiry. Any change to the subscription length will override the customers existing subscription.

Return type	Description
String	Subscription token

Throws	Description
AccountAPINGException	Generic exception that is thrown if this operation fails for any reason

*Since *

getApplicationSubscriptionToken

Operation

getApplicationSubscriptionToken

String getApplicationSubscriptionToken (int subscriptionLength **) throws AccountAPINGException**

Used to create new subscription tokens for an application. Returns the newly generated subscription token which can be provided to the end user.

Please note: A maximum number of **15,000** subscription **UNACTIVATED** subscriptions tokens can be created at any one time. Attempts to create more subscription tokens will return the error **TOO_MANY_REQUESTS** error which will restrict creation of further tokens until existing **UNACTIVATED** subscription tokens have been **ACTIVATED** or **CANCELLED**

Parameter name	Type	Required	Description
subscriptionLength	int		How many days the subscription should last. Open ended if value not supplied. Expiry time will be rounded up to midnight on the date of expiry.
clientReference	String		Any client reference for this subscription token request.

Return type	Description
String	Subscription token

Throws	Description
AccountAPINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

getApplicationSubscriptionHistory

getApplicationSubscriptionHistory

List<SubscriptionHistory> getApplicationSubscriptionHistory (String vendorClientId) throws AccountAPINGException

Returns a list of subscriptions tokens that have been associated with the customers account. This allows a vendor to identify if a customer has a previous subscription to their application and the status of each subscription.

Parameter name	Type	Required	Description
vendorClientId	String		The unique customer identifier
applicationKey	String		The unique application identifier

Return type	Description
List<SubscriptionHistory>	List of subscription tokens associated with the account

Throws	Description
AccountAPINGException	Generic exception that is thrown if this operation fails for any reason.

*Since *

listAccountSubscriptionTokens

Operation

listAccountSubscriptionTokens

List< AccountSubscription > listAccountSubscriptionTokens ( ) throws AccountAPINGException

List of subscription tokens associated with the account

Return type	Description
List< AccountSubscription >	List of subscription tokens associated with the account

Throws	Description
AccountAPINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

listApplicationSubscriptionTokens

Operation

listApplicationSubscriptionTokens

List< ApplicationSubscription > listApplicationSubscriptionTokens (SubscriptionStatus subscriptionStatus **) throws AccountAPINGException**

Returns a list of subscription tokens for an application based on the subscription status passed in the request. Returns all subscription token details, including the client reference and vendor client Id associated with the subscription token.

Parameter name	Type	Required	Description
subscriptionStatus	SubscriptionStatus		Optionally filter response by Subscription status of the token

Return type	Description
List< ApplicationSubscription >	List of subscription tokens for an application

Throws	Description
AccountAPINGException	Generic exception that is thrown if this operation fails for any reason.

Since 1.0.0

isAccountSubscribedToWebApp

isAccountSubscribedToWebApp

boolean isAccountSubscribedToWebApp (String vendorId) throws AccountAPINGException

Return whether an account has authorised a web app.

Parameter name	Type	Required	Description
vendorId	String	✓	The id of the vendor to check subscription for

Return type	Description
boolean	Subscribed or not

Throws	Description
AccountAPINGException	

Since 1.0.0

token

token

VendorAccessTokenInfo token (String client_id, **GrantType grant_type**, String code, **String client_secret**, String refresh_token **) throws AccountAPINGException**

Generate web vendor session based on a standard session identifiable by auth code, vendor secret and app key

Parameter name	Type	Required	Description
client_id	String	✓	The vendor's vendorId
grant_type	GrantType	✓	Whether the vendor is using an authorisation code or a refresh token to get a session
code	String		The authorisation code used to lookup the session to be returned
client_secret	String	✓	The vendor's private key used to verify their identity
refresh_token	String		The vendor's refresh token if the grant_type is refresh_token

Return type	Description
VendorAccessTokenInfo	Response object containing VendorAccessToken, RefreshToken and optionally a Subscription Token if one was created

Throws	Description
AccountAPINGException	

Since 1.0.0

getVendorDetails

getVendorDetails

VendorDetails getVendorDetails (String vendorId) throws AccountAPINGException

Return details about a vendor from its identifier. Response includes Vendor Name and URL.

Parameter name	Type	Required	Description
vendorId	String	✓	The vendor's public identifier

Return type	Description
VendorDetails	Response object containing the vendor and the redirect url

Throws	Description
AccountAPINGException	

Since 1.0.0

revokeAccessToWebApp

revokeAccessToWebApp

Status revokeAccessToWebApp (long vendorId) throws AccountAPINGException

Remove the link between an account and a vendor web app. This will remove the refreshToken for this user-vendor pair subscription.

Parameter name	Type	Required	Description
vendorId	long	✓	The id of the vendor to revoke access for

Return type	Description
Status	Returns whether the request was successful or not

Throws	Description
AccountAPINGException	

Since 1.0.0

getAffiliateRelation

getAffiliateRelation

List<AffiliateRelation> getAffiliateRelation (List<String> vendorClientIds) throws AccountAPINGException

Return relation between a list of users and an affiliate

Parameter name	Type	Required	Description
vendorClientIds	List<String>	✓	List of client ids to check affiliation on

Return type	Description
List<AffiliateRelation>	List of affiliate relation status per user

Throws	Description
AccountAPINGException	

Since 1.0.0

Interface Definition Documents

The below documents provide a machine readable interface description for the Betfair Exchange API in XML format.

SportsAPING.xml

AccountAPING.xml

Heartbea...PING.xml

Exchange Stream API

Overview

The Exchange Streaming API provides low latency access to Betfair Exchange market data allowing you to subscribe to and efficiently track changes to market, price and order data.

The protocol is based on ssl sockets (normal) with a CRLF json protocol. We publish a definition of the schema of the json messages in the Swagger format.

We maintain sample code in Java, C# and Node.js here: https://github.com/betfair/stream-api-sample-code

Sample Application - C#, Java & Node.js

A console based C#, Java and Node.js sample application is available for the Market & Order Streaming API and is available via https://github.com/betfair/stream-api-sample-code

Users wishing to interact with the Streaming API using one of these languages are strongly advised to make use of this sample code.

Swagger Definition

For users wishing to use other languages or develop their own implementation, we provide a swagger schema to allow browsing & code generation.

We recommend using Swagger Code Gen (http://swagger.io/swagger-codegen/) for generation,

- As a pre-requisite Java version 7 or higher must be installed
- Download both:
  - The Swagger Code Gen jar from: https://oss.sonatype.org/content/repositories/releases/io/swagger/swagger-codegen-cli/2.2.1/swagger-codegen-cli-2.2.1.jar
  - The Swagger Definition from our GitHub repository: https://github.com/betfair/stream-api-sample-code/blob/master/ESASwaggerSchema.json
- Run the following command to view a list of available languages to generate code for: *java -jar swagger-codegen-cli-2.2.1.jar*

- Run the following command to generate the code: *java -jar swagger-codegen-cli-2.2.1.jar generate -i ESASwaggerSchema.json -l <LANGUAGE> -o <OUTPUT_DIRECTORY>*

The Swagger editor can also be used to view the domain model

- Use File -> Import File and choose the Swagger Definition downloaded from our **GitHub repository**

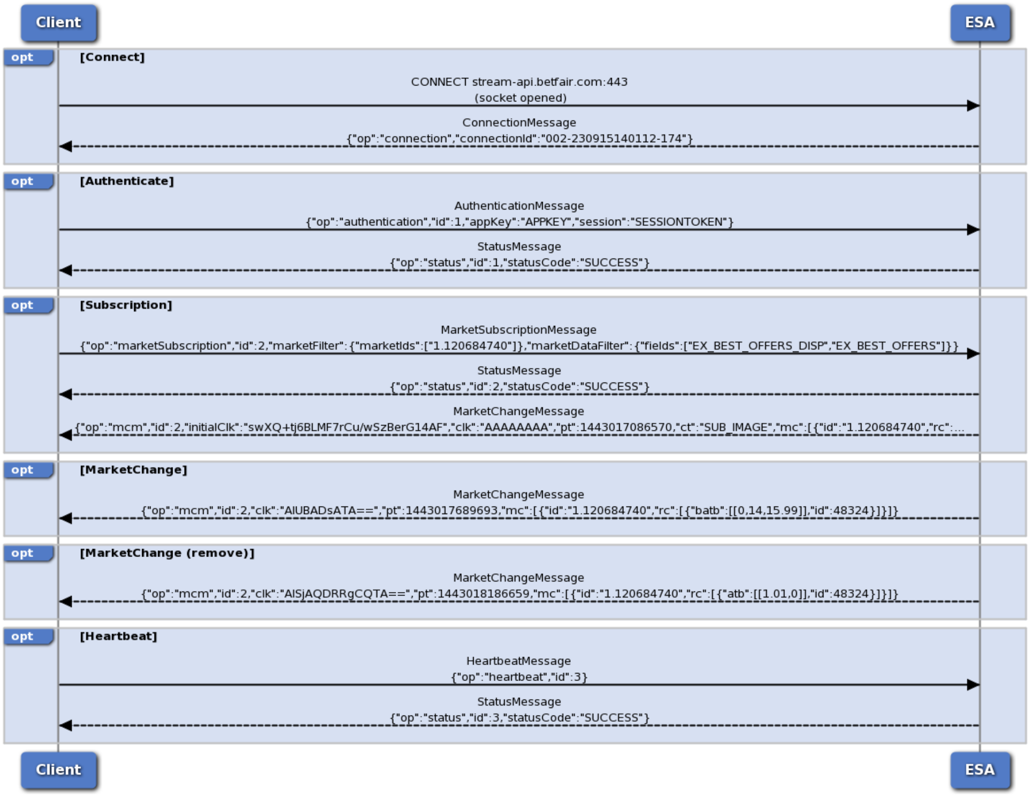
A few points to note with Swagger:

- It's cross platform and we can't control how it works / behaves - but it does save a lot of error prone typing.
- Enums and Inheritance are a little flaky:
    - Enums for error codes / filters etc. are defined but are treated as strings in c# (so you will need to copy definitions from the swagger spec until this is fixed by swagger).
    - Inheritance is defined but not generated correctly - you will have to manually manipulate the op=<type> field
        - In c# JsonCreationConverter is the typical way to model inheritance
        - In java look at JsonSubTypes
- We are not a REST service - so only the swagger generated model package is relevant.

Typical Interactions with Stream API:

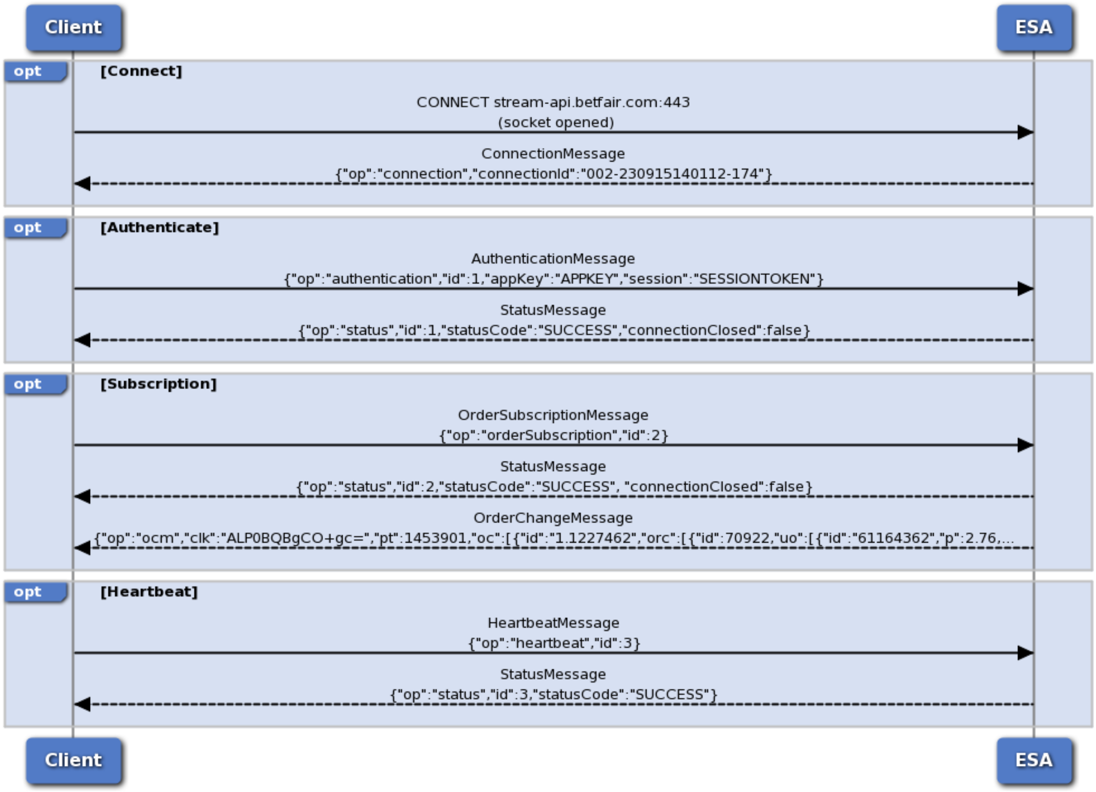
The typical API interactions are documented below (detail is below this).

Market Stream:



```
CONNECT stream-api.betfair.com:443
(socket opened)

ConnectionMessage
{"op":"connection","connectionId":"002-230915140112-174"}

AuthenticationMessage
{"op":"authentication","id":1,"appKey":"APPKEY","session":"SESSIONTOKEN"}

StatusMessage
{"op":"status","id":1,"statusCode":"SUCCESS"}

MarketSubscriptionMessage
{"op":"marketSubscription","id":2,"marketFilter":{"marketIds":["1.120684740"]},"marketDataFilter":{"fields":["EX_BEST_OFFERS_DISP","EX_BEST_OFFERS"]}}

StatusMessage
{"op":"status","id":2,"statusCode":"SUCCESS"}

MarketChangeMessage
{"op":"mcm","id":2,"initialClk":"swXQ+tj6BLMF7rCu/wSzBerG14AF","clk":"AAAAAAAA","pt":1443017086570,"ct":"SUB_IMAGE","mc":[{"id":"1.120684740","rc":...

MarketChangeMessage
{"op":"mcm","id":2,"clk":"AIUBADsATA==","pt":1443017689693,"mc":[{"id":"1.120684740","rc":[{"batb":[[0,14,15.99]],"id":48324}]}]}

MarketChangeMessage
{"op":"mcm","id":2,"clk":"AISjAQDRRgCQTA==","pt":1443018186659,"mc":[{"id":"1.120684740","rc":[{"atb":[[1.01,0]],"id":48324}]}]}

HeartbeatMessage
{"op":"heartbeat","id":3}

StatusMessage
{"op":"status","id":3,"statusCode":"SUCCESS"}
```

Order Stream:



CONNECT stream-api.betfair.com:443
(socket opened)

ConnectionMessage
{"op":"connection","connectionId":"002-230915140112-174"}

AuthenticationMessage
{"op":"authentication","id":1,"appKey":"APPKEY","session":"SESSIONTOKEN"}

StatusMessage
{"op":"status","id":1,"statusCode":"SUCCESS","connectionClosed":false}

OrderSubscriptionMessage
{"op":"orderSubscription","id":2}

StatusMessage
{"op":"status","id":2,"statusCode":"SUCCESS", "connectionClosed":false}

OrderChangeMessage
{"op":"ocm","clk":"ALP0BQBgCO+gc=","pt":1453901,"oc":[{"id":"1.1227462","orc":[{"id":70922,"uo":[{"id":"61164362","p":2.76,...

HeartbeatMessage
{"op":"heartbeat","id":3}

StatusMessage
{"op":"status","id":3,"statusCode":"SUCCESS"}

Connection

Protocol

Every message is in json & terminated with a line feed (CRLF):

```
{json message}\r\n
```

Json Serializer Setup

As the protocol is CRLF delimited don't forget to turn-off Json pretty printing (C# has this on by default)

TCP / SSL Connection

Connection is established with an SSL socket to the following address:

External (SSL):
`stream-api.betfair.com:443`

Avoiding TIMEOUT on connection

Once you have established a connection you should send a message within 15 seconds to avoid receiving a TIMEOUT error

Pre-production (beta) endpoint

For **pre-production (beta)** releases the following URL should be **used for integration testing only.**

Integration Endpoint
`stream-api-integration.betfair.com`

Basic Message Protocol

Two base message classes exist:

- RequestMessage - These are messages sent to the server.
- ResponseMessage - These are messages received from the server.

Every child message type has:

- id - A unique counter you should supply on a RequestMessage and which will be supplied back on a ResponseMessage.
- op - This identifies the request type and may be used to switch / deserialize correctly

Note: Any fields representing time and having a long type will represent the UNIX Timestamps (See https://currentmillis.com/ for conversions)

RequestMessage

RequestMessage is the base class for requests from the client; the discriminator is op=<message type>

Key fields:

- op=authentication - The AuthenticationMessage - authenticates your connection.
- op=marketSubscription - The MarketSubscriptionMessage - subscribes to market changes.
- op=orderSubscription - The OrderSubscriptionMessage - subscribes to order changes.
- op=heartbeat - The HeartbeatMessage - use if you need to keep a firewall open or want to test connectivity.

RequestMessages

- Remember to set op=<message type> - otherwise we can't decode the request
- Remember to set id=<unique sequence> - this will let you link requests with responses (these should be logged and provided on support calls)
- Every RequestMessage will receive a StatusMessage with the status of the call (linked by the id that you send).
    - **All errors apart from SUBSCRIPTION_LIMIT_EXCEEDED close the connection**

ResponseMessage

ResponseMessage is the base class for responses back to the client; the discriminator is op=<message type>

Key fields:

- op=connection - The ConnectionMessage sent on your connection.
- op=status - The StatusMessage (returned in response to every RequestMessage)
- op=mcm - The MarketChangeMessage that carries the initial image and updates to markets that you have subscribed to.
- op=ocm - The OrderChangeMessage that carries the initial image and updates to orders that you have subscribed to.

ResponseMessages

As mentioned earlier the id=<request id> and links your request with your response.

ChangeMessages carry the id of the original request that established the subscription

Status / StatusMessage

Every request receives a status response with a matching id.

Key fields:

- statusCode - The status of the request i.e success / fail
    - SUCCESS - Call processed correctly
    - FAILURE - Call failed (inspect errorCode and errorMessage for reason)
- connectionClosed - Boolean set to true if the connection was closed as a result of a failure
- errorCode - The type of error in case of a failure - see the swagger spec / enum.
- errorMessage - Additional message in case of a failure
- connectionsAvailable – The number of additional connections you can open (populated only in response to authentication requests)

ErrorCode

This categorizes the various error codes that could be expected (these are subject to change and extension)

Category	ErrorCode	Description

Protocol		**General errors not sent with id linking to specific request (as no request context)**
	INVALID_INPUT	`Failure code returned when an invalid input is provided (could not deserialize the message)`
	TIMEOUT	`Failure code when a client times out (i.e. too slow sending data)`
Authentication		**Specific to authentication**
	NO_APP_KEY	Failure code returned when an application key is not found in the message
	INVALID_APP_KEY	Failure code returned when an invalid application key is received
	NO_SESSION	Failure code returned when a session token is not found in the message
	INVALID_SESSION_INFORMATION	Failure code returned when an invalid session token is received
	NOT_AUTHORIZED	Failure code returned when client is not authorized to perform the operation
	MAX_CONNECTION_LIMIT_EXCEEDED	Failure code returned when a client tries to create more connections than allowed to
	TOO_MANY_REQUESTS	Failure code returned when a client makes too many requests within a short time period
Subscription		**Specific to subscription requests**
	SUBSCRIPTION_LIMIT_EXCEEDED	Customer tried to subscribe to more markets than allowed to - set to 200 markets by default
	INVALID_CLOCK	Failure code returned when an invalid clock is provided on re-subscription (check initialClk / clk supplied)
General		**General errors which may or may not be linked to specific request id**
	UNEXPECTED_ERROR	Failure code returned when an internal error occurred on the server
	CONNECTION_FAILED	Failure code used when the client / server connection is terminated

Connection / ConnectionMessage

This is received by the client when it successfully opens a connection to the server

Key fields:

- connectionId - This is a unique identifier that **you must supply for support.**

Initial ConnectionMessage

On establishing a connection a client receives a ConnectionMessage - the **connectionId must be logged & supplied on any support queries**:

```
{"op":"connection","connectionId":"002-230915140112-174"}
```

Authentication / AuthenticationMessage

This message is the first message that the client must send on connecting to the server - you must be authenticated before any other request is processed.

Key fields:

- op=authentication - This is the operation type
- appKey - This is your application key to identify your application
- session - The session token generated from API login.

Common Authentication Errors

Some common authentication errors that you should handle - these are defined on ErrorCodes enum (these will all close your connection):

- NO_APP_KEY / INVALID_APP_KEY - Check you are using the correct app key

- NO_SESSION / INVALID_SESSION_INFORMATION - Check the session is current
- NOT_AUTHORIZED - Check that you are using the correct appkey / session and that it has been setup by BDP
- MAX_CONNECTION_LIMIT_EXCEEDED - Check that you are not creating too many connections / are closing connections properly.
- TOO_MANY_REQUESTS – Check that you are not creating/closing connections too frequently

Subscription / SubscriptionMessage

This message changes the client's subscription - there are currently two subscription message types:

- op=marketSubscription- MarketSubscriptionMessage which streams:
    - op=mcm - MarketChangeMessage - the price changes for a market
- op=orderSubscription- OrderSubscriptionMessage which streams:
    - op=ocm - OrderChangeMessage - the order changes for a market

On creating a subscription you will receive:

- StatusMessage confirming the status of your request
- A stream of ChangeMessages linked with the id of the request which is composed of:
    - Initial image
    - Deltas to the initial image

It is possible to subscribe multiple times - each replaces the previous (each will send a new initial image and deltas) - they are not additive.

Key fields on a SubscriptionMessage:

- segmentationEnabled=true
    - segmentation breaks up large messages and improves: end to end performance, latency, time to first and last byte
    - see the topic on change message segmentation for a full explanation of how this works.
- conflateMs - Specifies a forced conflation rate (in milliseconds) - **Please note**: the field value will be 180000 if you access the Stream API using a Delayed App Key or have an account delay in place when using the Live App Key.
- heartbeatMs - Specifies a minimum interval that a client would expect to receive a message (in milliseconds) - bounds are 500 to 5000 milliseconds.
    - If no change is delivered in this interval then an empty change message will be sent with a ChangeType.HEARTBEAT
- initialClk & clk - these two sequence tokens allow for faster recovery in the event of a disconnection:
    - If supplied (with identical subscription criteria) you will receive a delta to your previous state rather than a full initial image
    - see the topic on re-subscription for a full explanation of how this works.

ChangeMessage

This message is the payload that delivers changes (both initial image & updates) to a client - there are currently two change message types:

- op=mcm - MarketChangeMessage
- op=ocm - OrderChangeMessage

The Order Changes and Market Changes are being produced by 2 independent systems so we can give no guarantee as to the order in which they will be sent.

Key fields on a ChangeMessage:

- ct= ChangeType - this enumeration is used to identify the type of change
    - SUB_IMAGE - The initial image returned from a subscribe. May also happen while subscription is on-going and should replace local cache entirely.
    - RESUB_DELTA - A patch returned from a resubscribe
    - HEARTBEAT - An empty message published if no data has been sent within heartbeatMs
        - We send these to maintain the connection to you and detect closed connections
        - You can use the heartbeatMs to verify that you are still connected
    - <null / not set> - An update message
- segmentType - SegmentType - this enumeration identifies multi-part segmented messages:
    - SEG_START - Start of a segmented message
    - SEG - Middle part of a segmented message
    - SEG_END - Last part of a segmented message
    - <null / not set> - A non-segmented message
- conflateMs - the actual conflation being used
    - This might be different to what you specified - if you account is for instance delayed or your request was out of bounds
- status - Stream status: set to null if the exchange stream data is up to date and **503** if the downstream services are experiencing latency
- heartbeatMs - the actual heartbeat being used
    - This might be different to what you specified as we bounds check
    - You can use this to verify your connection is live (as you should receive 1 message within this time period).
- pt - publishTime - the time we sent the message
- initialClk & clk - these two sequence tokens allow for faster recovery in the event of a disconnection:
    - If we send these then they should be stored
    - see the topic on re-subscription for a full explanation of how this works.

Heartbeat ChangeMessages

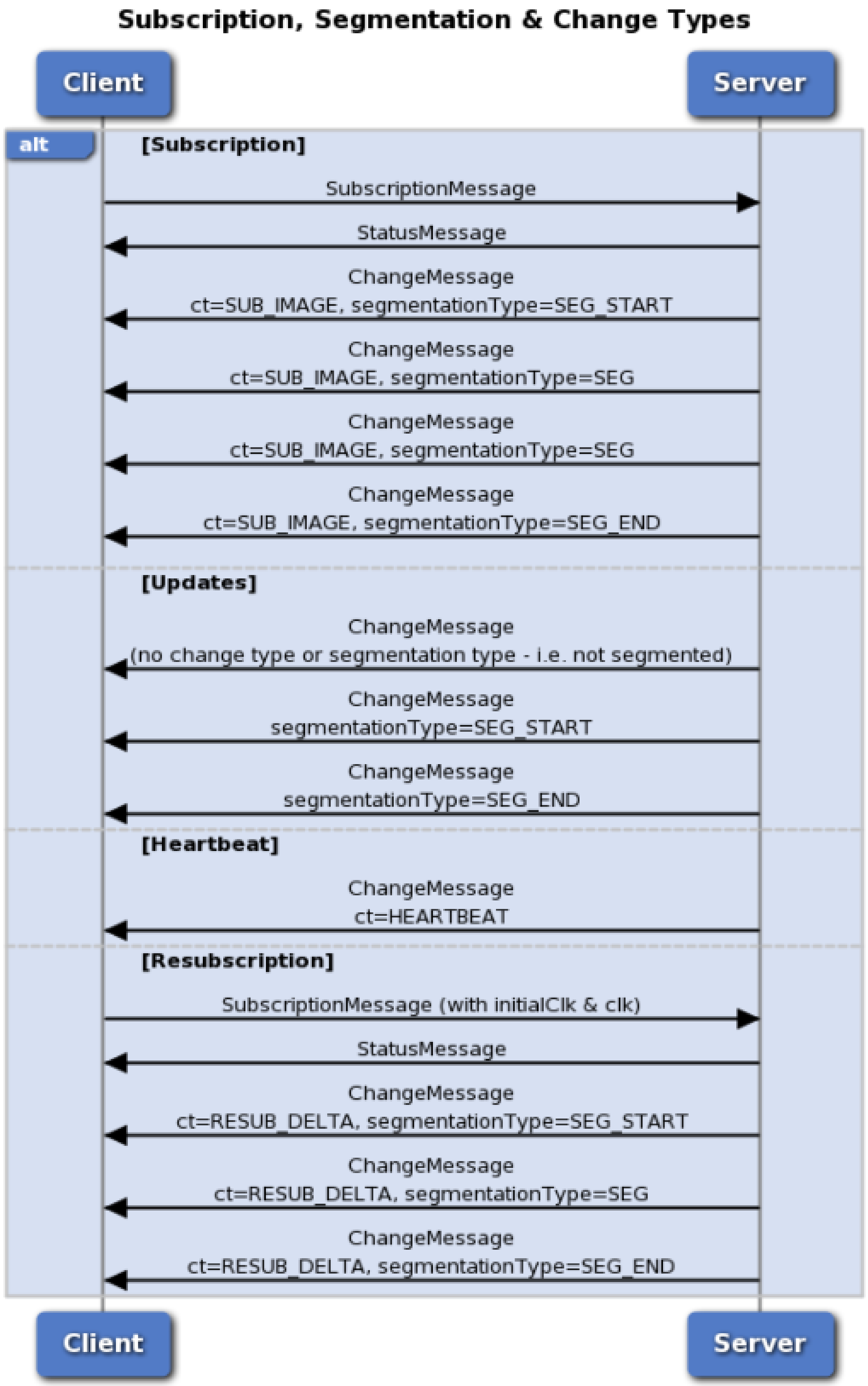
heartbeatMs is a guarantee of how often (even with no changes) you will receive a ChangeMessage; i.e.:

If heartbeatMs= 500 and your subscription has not changed in 500ms then we will send an empty ChangeMessage with ct=HEARTBEAT

(this verifies your connection is live and processing data)

Change Message Segmentation

The below shows the key interactions for subscription & changes with segmentation applied:


Subscription, Segmentation & Change Types
Client
Server
alt
[Subscription]
SubscriptionMessage
StatusMessage
ChangeMessage
ct=SUB_IMAGE, segmentationType=SEG_START
ChangeMessage
ct=SUB_IMAGE, segmentationType=SEG
ChangeMessage
ct=SUB_IMAGE, segmentationType=SEG
ChangeMessage
ct=SUB_IMAGE, segmentationType=SEG_END
[Updates]
ChangeMessage
(no change type or segmentation type - i.e. not segmented)
ChangeMessage
segmentationType=SEG_START
ChangeMessage
segmentationType=SEG_END
[Heartbeat]
ChangeMessage
ct=HEARTBEAT
[Resubscription]
SubscriptionMessage (with initialClk & clk)
StatusMessage
ChangeMessage
ct=RESUB_DELTA, segmentationType=SEG_START
ChangeMessage
ct=RESUB_DELTA, segmentationType=SEG
ChangeMessage
ct=RESUB_DELTA, segmentationType=SEG_END
Client
Server

Typically on changing your subscription you will want to clear any local cache you maintain.

Initial Image Handling

✓

- How can I detect the start of an initial image & clear my cache?

    - ct=ChangeType.SUB_IMAGE and segmentType=null or SegmentType.SEG_START indicates the start of a new image
- How can I detect the end of an initial image?
    - ct=ChangeType.SUB_IMAGE and segmentType=null or SegmentType.SEG_END indicates the end of a new image
- When I change Subscription how do I safely ignore messages for a previous subscription?
    - All ChangeMessages carry have id=<request id> this allows safe disposal during subscription change

MarketSubscriptionMessage

This subscription type is used to receive price changes for one or more markets; your subscription criteria determine what you see.

Coarse vs Fine Grain Subscriptions

It is preferable to use coarse grain subscriptions (subscribe to a super-set) rather than fine grain (specific market ids).

- If you find yourself frequently changing subscriptions you probably want to find a wider super-set to subscribe to

A MarketSubscription has two types of filter:

- marketFilter - MarketFilter - this is a horizontal filter of markets that you require (i.e. rows)
- marketDataFilter - MarketDataFilter - this is a vertical filter of fields that you require (i.e. columns)

Limiting the amount of data that you consume will make your initial image much smaller (and faster) & suppress changes that are uninteresting to you.

Market Filtering / MarketFilter

As with the APING API users have the ability to filter the market data they get from the new Exchange Stream API (ESA).

All subscriptions are evaluated with a few default criteria:

- Standard jurisdictional filtering that restricts visibility (mirroring site behavior)
- Permissions that control:
    - Specific sports that you are entitled for
    - A maximum consumption limit (exceeding this will result in an error with details of the limit: ErrorCode. SUBSCRIPTION_LIMIT_EXCEEDED)

Users can then specify the following filters when they subscribe to ESA:

Filter name	Type	Mandatory	Description
marketIds	Set<String>	No	If no marketIds passed user will be subscribed to all markets
bspMarket	Boolean	No	Restrict to bsp markets only, if True or non-bsp markets if False. If not specified then returns both BSP and non-BSP markets
bettingTypes	Set<BettingType>	No	Restrict to markets that match the betting type of the market (i.e. Odds, Asian Handicap Singles, or Asian Handicap Doubles)
eventTypeIds	Set<String>	No	Restrict markets by event type associated with the market. (i.e., "1" for Football, "7" for Horse Racing, etc)
eventIds	Set<String>	No	Restrict markets by the event id associated with the market.
turnInPlayEnabled	Boolean	No	Restrict to markets that will turn in play if True or will not turn in play if false. If not specified, returns both.
marketTypes	Set<String>	No	Restrict to markets that match the type of the market (i.e., MATCH_ODDS, HALF_TIME_SCORE). You should use this instead of relying on the market name as the market type codes are the same in all locales
venues	Set<String>	No	Restrict markets by the venue associated with the market. Currently only Horse Racing markets have venues.
countryCodes	Set<String>	No	Restrict to markets that are in the specified country or countries
raceTypes	Set<String>	No	Restrict to markets of a specific raceType. Valid values are - Harness, Flat,Hurdle, Chase, Bumper, NH Flat, NO_VALUE

Example marketSubscription

For example a subscription message with almost all filters enabled will look something like this:

```
{"op":"marketSubscription","id":2,"marketFilter":{"marketIds":["1.120684740"],"bspMarket":true,"bettingTypes":
["ODDS"],"eventTypeIds":["1"],"eventIds":["27540841"],"turnInPlayEnabled":true,"marketTypes":["MATCH_ODDS"],"
countryCodes":["ES"]},"marketDataFilter":{}}
```

✓ Subscriptions with no matching markets

We don't verify your subscription criteria as you could potentially subscribe to either a wild card (which would include future markets) or a future marketid which we do not have yet but would send on arrival

Market data field filtering / MarketDataFilter

A market data filter restricts the fields that you get back (and only if the fields have changed).

Key fields:

- fields - A set of field filter flags (see below)
- ladderLevels - For depth based ladders the number of levels to send (1 to 10). 1 is best price to back or lay etc.

Ladder Levels Behaviour

✓ When bdatb and bdatl are sent with an empty array ("bdatb:[]"), this indicates that there's an update but this has been filtered out due to the "ladderLevels" marketDataFilter i.e. the update falls outside of the "ladderLevels" specified.

The field filter flags are defined as:

Filter name	Fields:	Type	Description
EX_BEST_OFFERS_DISP	bdatb, bdatl	level, price, size	Best prices including Virtual Bets - depth is controlled by ladderLevels (1 to 10) - **Please note:** The virtual price stream is updated ~150 m/s after non-virtual prices. Virtual prices are calculated for all ladder levels.
EX_BEST_OFFERS	batb, batl	level, price, size	Best prices not including Virtual Bets - depth is controlled by ladderLevels (1 to 10)
EX_ALL_OFFERS	atb, atl	price, size	Full available to BACK/LAY ladder
EX_TRADED	trd	price, size	Full traded ladder
EX_TRADED_VOL	tv	size	Market and runner level traded volume
EX_LTP	ltp	price	Last traded price
EX_MARKET_DEF	marketDefinition	MarketDefinition	Send market definitions.
SP_TRADED	spb, spl	price, size	Starting price ladder
SP_PROJECTED	spn, spf	price	Starting price projection prices

Examples

Multiple field filters may be combined; a subscription message that contains data fields should look like the following:

```
{"op":"marketSubscription","id":2,"marketFilter":{"marketIds":["1.120684740"]},"marketDataFilter":{"fields":
["EX_BEST_OFFERS_DISP","EX_BEST_OFFERS","EX_ALL_OFFERS","EX_TRADED","EX_TRADED_VOL","EX_LTP","EX_MARKET_DEF","
SP_TRADED","SP_PROJECTED"]}}
```

The below example shows how to correctly use the **ladderLevels** marketDataFilter:

{"op": "marketSubscription", "id": 1, "marketFilter": { "marketIds": ["1.134085859"] }, "marketDataFilter": { "ladderLevels": 2, "fields": ["EX_MARKET_DEF", "EX_BEST_OFFERS"] } }

Correctly configuring field filters

✓ Correctly configuring field filters can help by:

- Reducing the size (and time) of initial images
- Reducing the rate of change (as only changes matching your field filter are sent)

MC / MarketChangeMessage

This is the ChangeMessage stream of data we send back to you once you subscribe to the market stream.

Key fields:

- <as for ChangeMessage>
- mc / MarketChange - this list of market changes contains the changes the markets that you have subscribed to.
    - img / Image - replace existing prices / data with the data supplied: it is not a delta (or null if delta)
    - marketDefinition / MarketDefinition - this is sent in full (but only if it has changed)
    - rc / RunnerChange - this is sent to supply the details of a runner (namely prices)
        - con / Conflated = true - if this is sent then more than one change is combined in this message (purely informational).
        - Values - **Please note:** these are only sent if the value has changed.
            - tv - Traded Volume
            - ltp - Last Traded Price
            - spn - Starting Price Near
            - spf - Starting Price Far
        - Level / Depth Based Ladders (level, price, size - triples - keyed by level):
            - size=0 - indicates a remove
            - batb / batl - Best Available To Back / Best Available To Lay (non-virtual)
            - bdatb / bdatl - Best Display Available To Back / Best Display Available To Lay (virtual)
        - Price point / full depth Ladders (price, size - tuples - keyed by price):
            - size=0 - indicates a remove
            - atb / atl - Available To Back / Available To Lay (these are the raw / full depth non-virtual prices)
            - spb / spl - Starting Price (Available To) Back / Starting Price (Available To) Lay (please be aware that these values are aligned with atb / atl)
            - trd - Traded

Building a price cache

Most of the change based data (RunnerChange) is delta based - this means a few rules:

- img / Image - if this is set to true then you should replace this item in your cache
- Values - the values sent are nullable & are not sent if they are not changed (i.e. if tv has not changed then there will be no field in the message)
- Level / Depth Based ladders
    - [0, 1.2, 20] -> Insert / Update level 0 (top of book) with price 1.2 and size 20
    - [0, 1.2, 0] -> Remove level 0 (top of book) i.e. ladder is now empty
- Price point / full depth ladders
    - [1.2, 20] -> Insert / Update price 1.2 with size 20
    - [1.2, 0] -> Remove price 1.2 i.e. there is no size at this price

Examples

You will always receive an update at every position in the ladder that changed so you'll never have to assume anything based on the message you receive.

Seeing [position,0,0] means that there’s nothing at that position anymore (and hence [0,0,0] means there’s nothing in the entire ladder anymore)

- **Placed the first bet on a selection**

`"batl":[[0,1.4,2],[1,0,0],[2,0,0],[3,0,0],[4,0,0],[5,0,0],[6,0,0],[7,0,0],[8,0,0],[9,0,0]]`

- **Placed a second bet that didn't disturb the first bet's position**

`"batl":[[1,1.5,2]]`

- **Placed a third bet that bumped the previous two down the ladder**

`"batl":[[2,1.5,2],[1,1.4,2],[0,1.3,2]]`

- **Cancelled the top position causing the other positions to move up (and the bottom position to become empty)**

`"batl":[[2,0,0],[1,1.5,2],[0,1.4,2]]`

- **Cancelled by market to remove the remaining 2 positions in one go**

`"batl":[[1,0,0],[0,0,0]]`

OrderSubscriptionMessage

This subscription type is used to receive order changes; the subscription message has one type of filter

- orderFilter (optional)

OrderFilter

This optional filter already filters by your account; but additional data shaping is supported

Filter name	Type	Mandatory	Default	Description
accountIds	Set<Integer>	No	null	This is for internal use only & should not be set on your filter (your subscription is already locked to your account).
includeOverallPosition	Boolean	No	true	Returns overall / net position (OrderRunnerChange.mb / OrderRunnerChange.ml)
customerStrategyRefs	Set<String>	No	null	Restricts to specified customerStrategyRefs (specified in placeOrders) ; this will filter orders and StrategyMatchChanges accordingly (Note: overall position is not filtered)
partitionMatchedByStrategyRef	Boolean	No	false	Returns strategy positions (OrderRunnerChange.smc=Map<customerStrategyRef, StrategyMatchChange>) - these are sent in delta format as per overall position.

Example

{"op":"orderSubscription","orderFilter":{"includeOverallPosition":false,"customerStrategyRefs":["betstrategy1"],"partitionMatchedByStrategyRef":true},"segmentationEnabled":true}

OCM / OrderChangeMessage

This is the ChangeMessage stream of data we send back to you once you subscribe to the order stream.

Key fields:

- <as for ChangeMessage>
- oc / OrderAccountChange - the modifications to account's orders (will be null on a heartbeat)
    - closed - indicates when the market is closed
    - id / Market Id - the id of the market the order is on
    - orc / Order Changes - a list of changes to orders on a runner
        - fullImage - replace existing data with the data supplied: it is not a delta (or null if delta)
        - id / Selection Id - the id of the runner (selection)
        - hc / Handicap - the handicap of the runner (selection) (null if not applicable)
        - uo / Unmatched Orders - orders on this runner that are unmatched
            - Every order change is sent in full; the transient on a change to EXECUTION_COMPLETE is sent (but it would not be sent on initial image)
            - id / Bet Id - the id of the order
            - p / Price - the original placed price of the order
            - s / Size - the original placed size of the order
            - bsp / BSP Liability - the BSP liability of the order (null if the order is not a BSP order)
            - side / Side - the side of the order
            - status / Status - the status of the order (E = EXECUTABLE, EC = EXECUTION_COMPLETE)
            - pt / Persistence Type - whether the order will persist at in play or not (L = LAPSE, P = PERSIST, MOC = Market On Close)
            - ot / Order Type - the type of the order (L = LIMIT, MOC = MARKET_ON_CLOSE, LOC = LIMIT_ON_CLOSE)
            - pd / Placed Date - the date the order was placed
            - md / Matched Date - the date the order was matched (null if the order is not matched)
            - ld / Lapsed Date - the date the order was lapsed (null if the order is not lapsed)
            - lsrc/Lapse Status Reason Code - the reason that some or all of this order has been lapsed (null if no portion of the order is lapsed)
            - avp / Average Price Matched - the average price the order was matched at (null if the order is not matched
            - sm / Size Matched - the amount of the order that has been matched
            - sr / Size Remaining - the amount of the order that is remaining unmatched
            - sl / Size Lapsed - the amount of the order that has been lapsed
            - sc / Size Cancelled - the amount of the order that has been cancelled
            - sv / Size Voided - the amount of the order that has been voided
            - rac / Regulator Auth Code - the auth code returned by the regulator
            - rc / Regulator Code - the regulator of the order
            - rfo / Reference Order - the customer supplied order reference
            - rfs / Reference Strategy - the customer supplied strategy reference used to group orders together - default is ""
        - Price point / full depth Ladders (price, size - tuples - keyed by price) of matches:
            - mb / Matched Backs - matched amounts by distinct matched price on the Back side for this runner
            - ml / Matched Lays - matched amounts by distinct matched price on the Lay side for this runner

Building an order cache

An order cache is somewhat simpler as orders are sent in full (on change) and only matches need delta merging

- fullImage - if this is set to true then you should replace this item in your cache
- Orders - replace each order according to order id.
- Price point / full depth ladders
    - [1.2, 20] -> Insert / Update price 1.2 with size 20
    - [1.2, 0] -> Remove price 1.2 i.e. there is no size at this price
    - An empty list of points also means the ladder is now empty

Currencies
✓

Market subscriptions - are always in underlying exchange currency - GBP. The default roll-up for GBP is £1 for **batb / batl** and **bdatb / bdatl**, This means that stakes of less than £1 (or currency equivalent) are rolled up to the next available price on the odds ladder. For **atb / atl** there is no roll-up. Available volume is displayed at all prices including those with less than £2 available.

Orders subscriptions - are provided in the currency of the account that the orders are placed in.
Unmatched Orders
✓
New subscriptions: Will receive an initial image with only E - Executable orders (unmatched).

Live subscriptions: Will receive a transient of the order to EC - Execution Complete as the order transits into that state (allowing you to remove the order from your cache).

Example Output of Order Stream Message on Connection/Re-connection

Here's an example showing the data provided following a connection/re-connection to the Order Stream API. The example shows matched backs on two separate markets one of which has a size remaining of 0.25.

Example of Order Stream Output (reconnection) - with size remaining

```
{
        "op": "ocm",
        "id": 6,
        "initialClk": "GpOH0JwBH762w50BHKKomJ0BGpzR5ZoBH5mWsJwB",
        "clk": "AAAAAAAAAAAAAA==",
        "conflateMs": 0,
        "heartbeatMs": 5000,
        "pt": 1468943673782,
        "ct": "SUB_IMAGE",
        "oc": [{
                "id": "1.125657695",
                "orc": [{
                        "fullImage": true,
                        "id": 48756,
                        "mb": [
                                [1.4, 2]
                        ]
                }]
        }, {
                "id": "1.125657760",
                "orc": [{
                        "fullImage": true,
                        "id": 151478,
                        "uo": [{
                                "id": "71352090695",
                                "p": 12,
                                "s": 5,
                                "side": "B",
                                "status": "E",
                                "pt": "L",
                                "ot": "L",
                                "pd": 1468919099000,
                                "md": 1468933833000,
                                "avp": 12,
                                "sm": 4.75,
                                "sr": 0.25,
                                "sl": 0,
                                "sc": 0,
                                "sv": 0
                        }],
                        "mb": [
                                [12, 4.75]
                        ]
                }]
        }]
}
```

Remaining 0.25 is then matched on marketId **1.125657760**

Example of Order Stream Output - with size remaining matched

```
{
        "op": "ocm",
        "id": 10,
        "initialClk": "GtD10ZwBH5OJxZ0BHK75mZ0BGsKq6JoBH4THsZwB",
        "clk": "AAAAAAAAAAAAAA==",
        "conflateMs": 0,
        "heartbeatMs": 5000,
        "pt": 1468944647413,
        "ct": "SUB_IMAGE",
        "oc": [{
                "id": "1.125670254",
                "orc": [{
                        "fullImage": true,
                        "id": 5643663
                }]
        }, {
                "id": "1.125657760",
                "orc": [{
                        "fullImage": true,
                        "id": 151478,
                        "mb": [
                                [12, 5]
                        ]
                }]
        }, {
                "id": "1.125657695",
                "orc": [{
                        "fullImage": true,
                        "id": 48756,
                        "mb": [
                                [1.4, 2]
                        ]
                }]
        }]
}
```

Heartbeat / HeartbeatMessage

This is an explicit heartbeat request (in addition to server heartbeat interval which is automatic).

This functionality should not normally be necessary unless you need to keep a firewall open.

Do I need to use HeartbeatMessage?

No - under normal circumstances the subscription level ChangeType.HEARTBEAT is an acceptable guarantee of connection health.

Use the HeartbeatMessage only if you need to keep a firewall open - as it will incur some performance penalty (as a response will block your connection)

Re-connection / Re-subscription

If a client is disconnected a client may connect, authenticate and re-subscribe.

Prerequisite steps:

- Store your subscription criteria (re-subscribe will only work correctly with identical subscription criteria.
- Store initialClk (normally only initial image) & Clk (normally on every non-segmented message or a SEG_END) on any change message they are sent on.

Connection is broken.

- Connect & Authenticate as normal
- Subscribe setting initialClk and Clk to the last values sent on the subscription
- Change message with ChangeType.RESUB_DELTA is sent - this will patch your cache
- Some markets might have img=true set indicating they are either new or can't be patched.

Easiest way to implement re-subscribe

- Store any new subscription message you send as a "pending subscription"
- Store this as a "active subscription" once you get your initial image
- Update the initialClk & clk on the subscription message with any non-null values
- Resend this message after re-connecting

Performance Considerations

Here are a few tips on performance which are worth bearing in mind:

Performance tips

- A single market subscription & a subscription to all markets have an identical latency:
  - Cost is identical as the two subscriptions above would evaluate in sequence and thus with the same average latency.
  - Initial image is more costly to send than extra updates.
  - Limiting data with appropriate filters reduces initial image time
- Segmented data will always out perform non-segmented data:
  - You will be processing a buffer while another is in-flight and another is being prepared to send
- Writes to your connection are directly effected by how quickly you consume data & clear your socket's buffer
  - Consuming data slowly is effectively identical to setting conflation.
  - If you receive con=true flag on a market - then you are consuming data slower than the rate of delivery.

Currency Support

The Exchange Stream API supports GBP currency only.

Those looking to convert data from GBP to a different currency should use listCurrencyRates to do so.

Currencies

Market subscriptions - are always in underlying exchange currency - GBP. The default roll-up for GBP is £1 for **batb / batl** and **bdatb / bdatl**, This means that stakes of less than £1 (or currency equivalent) are rolled up to the next available price on the odds ladder. For **atb / atl** there is no roll-up. Available volume is displayed at all prices including those with less than £2 available.

Orders subscriptions - are provided in the currency of the account that the orders are placed in.

Runner Removals on the Order Stream

When a Rule 4 Runner Removal occurs in a Horse Race the price of matched bets on remaining runners are reduced by a Reduction Factor.

For these matched bets, you will receive on the Order Stream both a uo for the affected bet and the relevant updates to mb or ml (reducing the matched volume at the original matched price and adding volume at the new reduced price).

Initial bet placement at price 12

```
{"op":"ocm","id":2,"clk":"AK0CAPsBALEC","pt":1467219304831,"oc":[{"id":"1.102151675","orc":[{"fullImage":true,"i
d":6113662,"uo":[{"id":"10822867886","p":12,"s":2,"side":"B","status":"E","pt":"L","ot":"L","pd":1467219304000,"
sm":0,"sr":2,"sl":0,"sc":0,"sv":0,"rac":"","rc":"REG_GGC"}]}]}]}
```

Bet fully matched at price 12

```
{"op":"ocm","id":2,"clk":"AK0CAPsBALMC","pt":1467219316709,"oc":[{"id":"1.102151675","orc":[{"id":6113662,"uo":
[{"id":"10822867886","p":12,"s":2,"side":"B","status":"EC","pt":"L","ot":"L","pd":1467219304000,"md":14672193160
00,"avp":12,"sm":2,"sr":0,"sl":0,"sc":0,"sv":0}],"mb":[[12,2]]}]}]}
```

Runner removed (and so bet reduced in price to 9.47)

```
{"op":"ocm","id":2,"clk":"AK0CAJACALsC","pt":1467219376611,"oc":[{"id":"1.102151675","orc":[{"id":6113662,"uo":
[{"id":"10822867886","p":12,"s":2,"side":"B","status":"EC","pt":"L","ot":"L","pd":1467219304000,"md":14672193160
00,"avp":9.47,"sm":2,"sr":0,"sl":0,"sc":0,"sv":0}],"mb":[[9.47,2],[12,0]]}]}]}
```

See the avp in the uo record showing the new price of 9.47 and see the two entries in mb, one to remove the previously added size of 2 at price point 12 and one to add the size of 2 into the new price point of size 9.47.

Bets placed on the actual removed runner will be voided/lapsed (for matched/unmatched bets respectively) and these will also be sent through on the Order Stream.

Line Markets

Line markets being sent on the Market Stream can be identified by the bettingType field of MarketDefinition (with value of "LINE").

The MarketDefinition of Line markets provide some additional fields that will be null for all other types,

- lineMaxUnit - maximum value for the outcome, in market units for this market (eg 100 runs).
- lineMinUnit - minimum value for the outcome, in market units for this market (eg 0 runs).
- lineInterval - the odds ladder on this market will be between the range of lineMinUnit and lineMaxUnit, in increments of the interval value.e.g. If lineMinUnit=10 runs, lineMaxUnit=20 runs, lineInterval=0.5 runs, then valid odds include 10, 10.5, 11, 11.5 up to 20 runs.

For updates for Orders on Line markets received on the Order Stream be aware of how the following properties behave,

- price - line markets operate at even-money odds of 2.0. However, price for these markets refers to the line positions available as defined by the markets min-max range and interval steps.
- side - for Line markets a 'B' bet refers to a SELL line and an 'L' bet refers to a BUY line.
- averagePriceMatched - this value is not meaningful for activity on Line markets and is not guaranteed to be returned or maintained for these markets.

Stream API Status - latency

Every **ChangeMessage**, for order and market stream, contains a '**status'** field which will give an indication on the health of the stream data provided by the service. This is feature will be used in addition to the heartbeat mechanism which only gives an indication that the service is up but doesn't provide an indication of the latency of the data provided.

By default, when the stream data is up to date the value is set to null and will be set to **503** when the stream data is unreliable (i.e. not all bets and markets changes will be reflected on the stream) due to an increase in push latency. Clients **shouldn't** disconnect if status 503 is returned; when the stream recovers updates will be sent containing the latest data. The status is sent per each subscription on heartbeats and change messages.

Example of message containing the status field:

```
{"op":"ocm","id":3,"clk":"AAAAAAAA","status":503,"pt":1498137379766,"ct":"HEARTBEAT"}

{"op":"mcm","id":2,"clk":"AAAAAAAA","status":503,"pt":1498137381621,"ct":"HEARTBEAT"
```

Stream Health

In addition to the Stream API status field we'd recommend the below as best practice for monitoring the health of the Stream API:

- Use heartbeat messages to confirm Stream API is healthy and that you are still connected
- Messages with ChangeType.HEARTBEAT will be sent at the requested interval if no change has occurred.
- If no message of any kind is received for 2 x heartbeat interval then you may no longer be connected so initiate a fresh connection (use re-subscribe to continue where you left off)
- Re-connect code should contain back offs to avoid spamming the service if you are unable to connect for a prolonged period for any reason
- if you receive con=true flag on a market - then you are consuming data slower than the rate of deliver. If the socket buffer is full we won't attempt to push; so the next push will be conflated.

Lapse Status Reason Code Possible Values

This field will now be present in some cases on the Order object of the Order Stream to denote the reason that some or all of the order is lapsed. It will be null if no portion of the order is lapsed or if the order lapsed for some reason other than those listed below.

The full list of currently supported values for this field is:

Code	Description
MKT_UNKNOWN	The market was unknown, presumably removed from the matcher (closed) between bet placement and matching.
MKT_INVALID	The market was known about, but in an invalid state.

RNR_UNKNOWN	The runner was unknown, presumably removed between bet placement and matching.
TIME_ELAPSED	The bet was waiting in the queue too long, so was lapsed for safety.
CURRENCY_UNKNOWN	The bet's currency ID was not recognised by the matcher.
PRICE_INVALID	The bet's price was invalid, e.g. outside the defined ladder for the market.
MKT_SUSPENDED	The market was suspended at the time the bet came to be matched.
MKT_VERSION	The bet had a maximum market version set, and the market's version on matching was greater than this.
LINE_TARGET	The bet was on a line market, but was requested targeting profit or payout.
LINE_SP	The bet was on a line market, but was either a BSP bet directly, or requested to PERSIST_TO_SP.
SP_IN_PLAY	The bet was a BSP bet that had somehow come to be placed after turn-in-play.
SMALL_STAKE	The bet's stake was worth less than half a penny in GBP.
PRICE_IMP_TOO_LARGE	When the bet came to be matched, the price available was better than its best permitted price, suggesting a significant shift in the market, presumably due to a major incident, which may have rendered the bet unwanted.

Offline Documentation

An offline version of the Exchange Stream API is available via ExchangeStreamAPI-March2018.pdf

Please note, the full Exchange Stream API specification is available online here

Known Issues

- **Markets moved under a new eventId** - In certain circumstances, a market may move from one eventId to another due to actions performed by our Exchange Operations team. This will cause the Exchange Stream API to hold two copies of the market in its cache and the initial image of the market provided will therefore contain both copies of the market. In these circumstances further Stream API updates will only be sent for the latest version of the market. You can identify the latest version of the market using the "version" parameter returned in the initial image and should only store the market with the higher version number.

Sample Code

As well as sample code developed by Betfair, this page allows you to find sample code or documentation prepared by members of the Developer Program community.

If you would like to contribute to this space, please contact Developer Support
All **Betfair** developed sample code follows a typical work flow:

- Find the next UK Horse Racing Win market
- Get prices for the market
- Place a bet on the market
- Handle the error returned by the API when the bet fails as it is below the minimum stake size.

Please Note: In order to aide ease of understanding, the basic Betfair samples are not intended to show certain best practices for speed and throughput. Well designed applications should follow the best practices for client design of the application/language platform and should optimise on an HTTP request level with features such as requesting gzip'd responses and http connection keep alives. Please see Optimizing API Application Performance for further details.

Sample Code

The following samples are currently available:

Language	Documentation	Available From	Description	Developed By
Java	Description	https://github.com/betfair/API-NG-sample-code/tree/master/java	Sample Code for Java	Betfair
Javascript	Description	https://github.com/betfair/API-NG-sample-code/tree/master/javascript	Sample Code for Node.js	Betfair
Python	Description	https://github.com/betfair/API-NG-sample-code/tree/master/python	Sample Code for Python	Betfair
PHP	Description	https://github.com/betfair/API-NG-sample-code/tree/master/php	Sample Code for PHP	Betfair
Excel/VBA	Description	https://github.com/betfair/API-NG-sample-code/tree/master/vba	Sample Code for Excel/VBA	Betfair
C#	Description	https://github.com/betfair/API-NG-sample-code/tree/master/cSharp	Sample Code for C#	Betfair
Curl	Description	https://github.com/betfair/API-NG-sample-code/tree/master/curl	Sample Curl Requests	Betfair
Perl		https://github.com/betfair/API-NG-sample-code/tree/master/perl	Sample Code for Perl	Betfair
Delphi		https://github.com/jamiei/Betfair-API-NG-Sample	Sample Code for Delphi	Community member - jamiei
Clojure		https://github.com/jamiei/betfair-aping-sample	Sample Code for Clojure	Community member - jamiei

Client Libraries & Sample Applications

Language	Available From	Description	Developed By
C#	https://github.com/joelpob/betfairng	API-NG Client Library for c#	Community member - joelpob
Java	https://github.com/joelpob/jbetfairng	Cleint library for Java	Community member - joelpob
Excel/VBA	https://github.com/betfair/API-NG-Excel-Toolkit	Excel Sample Spreadsheet Application	Robin Barrett
Delphi	https://github.com/betfair/API-NG-Delphi-Client	API-NG Client Library for Delphi	Community member - khughes
Javascript	https://github.com/AlgoTrader/betfair	API-NG Client Library for Node.js	Community member - Algotrader
Perl	https://github.com/MyrddinWyllt/WWW-BetfairNG	Perl Library for API-NG	Community member - Merlin
PHP	https://github.com/danieledangeli/betfair-php	API-NG Client Library for PHP	Community member - daniele8805
Ruby	https://github.com/mikecmpbll/betfair	Ruby wrapper for API-NG	Community member - mikecmpbll
Python	https://github.com/jmcarp/betfair.py	Python wrapper for API-NG	Community member - jmcarp
Python	https://github.com/liampauling/betfairlightweight	Lightweight python wrapper for Betfair API-NG (with streaming)	Community member - LiamP

Scala	https://github.com/city81/betfair-service-ng	Scala sample code for API-NG	Community member - theswan1
R	https://github.com/phillc73/abettor	Sample code for R	Community member - phill_c
C++	https://github.com/captain-igloo/greentop	C++ Betfair API Client	Community member - plachner
Rust	https://docs.rs/botfair/0.3.0/botfair/	Rust bindings for the Betfair API	Community member - esotericnonsense

Stream API

Language	Available From	Description	Developed By
C#	https://github.com/betfair/stream-api-sample-code/tree/master/csharp	Sample application for Stream API	Betfair
Java	https://github.com/betfair/stream-api-sample-code/tree/master/java	Sample application for Stream API	Betfair
Node.js	https://github.com/betfair/stream-api-sample-code/tree/master/node.js	Sample application for Stream API	Betfair

Historical Data

Code for processing the data provided by the Betfair Exchange historical data service available via https://historicdata.betfair.com

Language	Available From	Description	Developed By
Python	https://github.com/liampauling/betfairlightweight/tree/master/examples	Parse/output historical data allowing backtesting or with a custom listener, csv creation	Community member - LiamP
Excel/VBA	https://github.com/betfair/historic-data-workbook	Simple Excel workbook for use with BASIC historical data Coverts files into Excel and provides basic custom reporting tools	Robin Barrett

Tutorials

Language	Available From	Description	Developed By
R	https://betfair-datascientists.github.io/api/apiRtutorial/	Betfair API R Tutorial	Betfair Australia
Python	https://betfair-datascientists.github.io/api/apiPythontutorial/	Betfaur API Python Tutorial	Betfair Australia

Books

Language	Available From	Description	Author
Visual Basic	http://www.amazon.co.uk/Programming-Betfair-Creating-Trading-Applications/dp/151143211X	A Guide to Creating Trading Application for API-NG	James Butler

Optimizing API Application Performance

To optimize performance and ensure that your application is interacting with the Betfair API as efficiently as possible, we strongly recommend the following as best practice.

Expect: 100- Continue Header

Sending this header will result in the error: **"The remote server returned an error: (417) Expectation Failed."**

You should be aware that if using the .Net Framework you will need to set the relevant property in the ServicePointManager which then prevents the "Expect" header from being added:

System.Net.ServicePointManager.Expect100Continue = false;

Enabling HTTP compression

HTTP compression is a capability built into both web servers and web clients to reduce the number of bytes transmitted in an HTTP response. This makes better use of available bandwidth and increases performance while reducing download time. When enabled, HTTP protocol data is compressed before it is sent from the server. Clients capable of receiving compressed HTTP data announce that they support compression in the HTTP header. Almost all modern web browsers support HTTP Compression by default.

The Betfair API uses HTTP to handle communication between API clients and servers. Therefore, the JSON messages can be compressed using the same HTTP compression used by web browsers. Custom API applications may need some modification before they can take advantage of this feature. Specifically, they need to send an additional HTTP header to indicate they support receipt of compressed responses from the API. In addition, some environments require you to explicitly decompress the response.

We would therefore recommend that all Betfair API request are sent with the '**Accept-Encoding: gzip, deflate'** request header.

HTTP Persistent connection

We recommend that **Connection: keep-alive** header is set for all requests to guarantee a persistent connection and therefore reducing latency.

Other performance tips

Additional advice regarding optimizing HTTPClient performance can be found via http://hc.apache.org/httpclient-3.x/performance.html

C#

C-Sharp and API-NG

This page provides a guide on how to communicate with API-NG using C#, and some code snippets showing its purpose. The C-sharp code is written against .Net 4, and is a simple Console application.

In the sample code we use the Json-rpc and rescript protocol. All requests are sent and received using json format. To post a request we prepare the json string using C# object and then we serialize the object using the Json .Net library

This documentation refers to the code available at https://github.com/betfair/API-NG-sample-code/tree/master/cSharp

Expect- 100 Continue Header

Please note: When using .NET you set the "Expect -100Continue" property to false. This property sits in the ServicePointManager class

How To Run

Provided that you have .net 4 installed, and the .net related dll's sit in the location : C:\Program Files\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.0\Profile\Client\

The sample takes three arguments the app key and session token are mandatory. the third argument is to do with which client you want to use rescript or jsonrpc, if nothing is passed then its defaulted to jsonrpc.

Then it is a simple case of opening up command line and executing:

```
<Path To Cloned Repository>\Api-ng-sample-code\Api-ng-sample-code\bin\Release\Api-ng-sample-code.exe <appkey>
<sessontoken> <(optional)rescript/jsonrpc>
```

Code Snippet

Json-Rpc

This is the main, brain code snippet of the json-rpc calls all methods go through here, initially we create a request object, which contains necessary headers. the appkey and session token are in the custom headers which gets instantiated when the client is instantiated., the invoke methods actually serializes the request objects makes the request, and upon receiving the response, de-serialize it into the response object specified as the T using generics.

```
protected WebRequest CreateWebRequest(Uri uri)
      {
          WebRequest request = WebRequest.Create(new Uri(EndPoint));
          request.Method = "POST";
          request.ContentType = "application/json-rpc";
          request.Headers.Add(HttpRequestHeader.AcceptCharset, "ISO-8859-1,utf-8");
          request.Headers.Add(CustomHeaders);
          return request;
      }



      public T Invoke<T>(string method, IDictionary<string, object> args = null)
      {
          if (method == null)
              throw new ArgumentNullException("method");
          if (method.Length == 0)
              throw new ArgumentException(null, "method");

          var request = CreateWebRequest(new Uri(EndPoint));
```

```
            using (Stream stream = request.GetRequestStream())
            using (StreamWriter writer = new StreamWriter(stream, Encoding.UTF8))
            {
                var call = new JsonRequest { Method = method, Id = 1, Params = args };
                JsonConvert.Export(call, writer);
            }
            Console.WriteLine("Calling: " + method +  " With args: " + JsonConvert.
Serialize<IDictionary<string, object>>(args));

            using (WebResponse response = GetWebResponse(request))
            using (Stream stream = response.GetResponseStream())
            using (StreamReader reader = new StreamReader(stream, Encoding.UTF8))
            {
                var jsonResponse = JsonConvert.Import<T>(reader);
                Console.WriteLine("Got Response: " + JsonConvert.Serialize<JsonResponse<T>>(jsonResponse));
                if (jsonResponse.HasError)
                {
                    throw ReconstituteException(jsonResponse.Error);
                }
                else
                {
                    return jsonResponse.Result;
                }
            }
        }


        private static System.Exception ReconstituteException(Api_ng_sample_code.TO.Exception ex)
        {
            var data = ex.Data;

            // API-NG exception -- it must have "data" element to tell us which exception
            var exceptionName = data.Property("exceptionname").Value.ToString();
            var exceptionData = data.Property(exceptionName).Value.ToString();
            return JsonConvert.Deserialize<APINGException>(exceptionData);
        }
    }
        }
```

Example usage of the code above is:

```
        public IList<EventTypeResult> listEventTypes(MarketFilter marketFilter, string locale = null)
        {
            var args = new Dictionary<string, object>();
            args[FILTER] = marketFilter;
            args[LOCALE] = locale;
            return Invoke<List<EventTypeResult>>(LIST_EVENT_TYPES_METHOD, args);

        }
```

Rescript

Below is the rescript implementation of the functionality mentioned above

```
        protected HttpWebRequest CreateWebRequest(String restEndPoint)
        {
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(restEndPoint);
            request.Method = "POST";
            request.ContentType = "application/json";
            request.ContentLength = 0;
            request.Headers.Add(HttpRequestHeader.AcceptCharset, "UTF-8");
            request.Accept = "application/json";
            request.Headers.Add(CustomHeaders);
            return request;
        }

        public T Invoke<T>(string method, IDictionary<string, object> args = null)
        {
            if (method == null)
                throw new ArgumentNullException("method");
            if (method.Length == 0)
                throw new ArgumentException(null, "method");

            var restEndpoint = EndPoint + method + "/";
            var request = CreateWebRequest(restEndpoint);

            var postData = JsonConvert.Serialize<IDictionary<string, object>>(args) + "}";
            Console.WriteLine("Calling: " + method + " With args: " + postData);

            var bytes = Encoding.GetEncoding("UTF-8").GetBytes(postData);
            request.ContentLength = bytes.Length;

            using (Stream stream = request.GetRequestStream())
            {
                stream.Write(bytes, 0, bytes.Length);
            }
            using (WebResponse response = GetWebResponse(request))

            using (Stream stream = response.GetResponseStream())
            using (StreamReader reader = new StreamReader(stream, Encoding.UTF8))
            {
                var jsonResponse = reader.ReadToEnd();
                Console.WriteLine("Got response: " + jsonResponse);
                if (jsonResponse.Contains("exception")) {
                    throw ReconstituteException(JsonConvert.Deserialize<Api_ng_sample_code.TO.Exception>
(jsonResponse));
                }
                return JsonConvert.Deserialize<T>(jsonResponse);
            }
        }

        private static System.Exception ReconstituteException(Api_ng_sample_code.TO.Exception ex)
        {
            var data = ex.Detail;

            // API-NG exception -- it must have "data" element to tell us which exception
            var exceptionName = data.Property("exceptionname").Value.ToString();
            var exceptionData = data.Property(exceptionName).Value.ToString();
            return JsonConvert.Deserialize<APINGException>(exceptionData);

        }
```

and the example usage

```
public IList<EventTypeResult> listEventTypes(MarketFilter marketFilter, string locale = null)
        {
            var args = new Dictionary<string, object>();
            args[FILTER] = marketFilter;
            args[LOCALE] = locale;
            return Invoke<List<EventTypeResult>>(LIST_EVENT_TYPES_METHOD, args);

        }
```

Example usage of the code above

```
IClient client = null;
            string clientType = null;
            if (args.Length == 3)
            {
                clientType = args[2];
            }
            // if rescript has been passed as the third argument use it otherwise default to json client
            if (!string.IsNullOrEmpty(clientType) && clientType.Equals("rescript"))
            {
                Console.WriteLine("Using RescriptClient");
                client =  new RescriptClient(Url, appkey, sessionToken);
            }else
            {
                Console.WriteLine("Using JsonRpcClient");
                client = new JsonRpcClient(Url, appkey, sessionToken);
            }
            Console.WriteLine("\nBeginning sample run!\n");
            var marketFilter = new MarketFilter();
            marketFilter.TextQuery = "Horse Racing";

            var eventTypes = client.listEventTypes(marketFilter);
```

Excel & VBA Sample

API-NG Excel VBA Sample Code

Prerequisites

- Microsoft Excel 2007 or later

Installation

None required
Clone the repository at https://github.com/betfair/API-NG-sample-code/tree/master/vba

How to run

Open the Excel workbook. Obtain an app key and session token and enter them into sheet **Example** cells **B3** and **B4** respectively.

JSON-RPC

- Click **Clear**
- Click **Go (JSON-RPC)** button

RESCRIPT (JSON)

- Click **Clear**
- Click **Go (RESCRIPT)** button

Code Snippets

Calling API-NG

```
Function SendRequest(Url, AppKey, Session, Data) As String
    On Error GoTo ErrorHandler:
    Dim xhr: Set xhr = CreateObject("MSXML2.XMLHTTP")

    With xhr
        .Open "POST", Url & "/", False
        .setRequestHeader "X-Application", AppKey
        .setRequestHeader "Content-Type", "application/json"
        .setRequestHeader "Accept", "application/json"
    End With

    If Session <> "" Then
        xhr.setRequestHeader "X-Authentication", Session
    End If

    xhr.send Data
    SendRequest = xhr.responseText

    If xhr.Status <> 200 Then
        Err.Raise vbObjectError + 1000, "Util.SendRequest", "The call to API-NG was unsuccessful. Status code:
" & xhr.Status & " " & xhr.statusText & ". Response was: " & xhr.responseText
    End If

    Set xhr = Nothing
    Exit Function

ErrorHandler:
    HandleError
End Function
```

Calling API-NG via JSON-RPC

```
Dim Request: Request = "{""jsonrpc"": ""2.0"", ""method"": ""SportsAPING/v1.0/listEventTypes"", ""params"":
{""filter"":{}}, ""id"": 1}"
Dim Url: Url = "https://api.betfair.com/json-rpc/"

Dim ListEventTypesResponse As String: ListEventTypesResponse = SendRequest(Url, "your app key", "your session
token", Request)
```

Calling API-NG via RESCRIPT

```
Dim Request: Request = "{""filter"":{}}"
Dim Url: Url = "https://api.betfair.com/rest/v1.0/listEventTypes/"
Dim ListEventTypesResponse As String: ListEventTypesResponse = SendRequest(Url, "your app key", "your session
token", Request)
```

Ascertain the event type id for Horse Racing using listEventTypes

Common

```
Function GetListEventTypesRequestString() As String
    GetListEventTypesRequestString = "{""filter"":{}}"
End Function

Function GetEventTypeIdFromEventTypes(ByVal EventTypes As Object) As String
    GetEventTypeIdFromEventTypes = "0"

    Dim Index As Integer
    For Index = 1 To EventTypes.Count Step 1
        Dim EventType: Set EventType = EventTypes.Item(Index).Item("eventType")
        If EventType.Item("name") = "Horse Racing" Then
            GetEventTypeIdFromEventTypes = EventType.Item("id")
            Exit For
        End If
    Next
End Function
```

JSON-RPC

```
Dim Request: Request = MakeJsonRpcRequestString(ListEventTypesMethod, GetListEventTypesRequestString())
Dim ListEventTypesResponse As String: ListEventTypesResponse = SendRequest(GetJsonRpcUrl(), GetAppKey(), "",
Request)
Dim EventTypeResult: Set EventTypeResult = ParseJsonRpcResponseToCollection(ListEventTypesResponse)
Dim EventTypeId: EventTypeId = GetEventTypeIdFromEventTypes(EventTypeResult)
```

RESCRIPT

```
Public Const ListEventTypesMethod As String = "listEventTypes"
Dim Request: Request = MakeJsonRpcRequestString(ListEventTypesMethod, GetListEventTypesRequestString())
Dim ListEventTypesResponse As String: ListEventTypesResponse = SendRequest(GetRestUrl() + ListEventTypesMethod,
GetAppKey(), "", Request)
Dim EventTypeResult: Set EventTypeResult = ParseRestResponseToCollection(ListEventTypesResponse)
Dim EventTypeId: EventTypeId = GetEventTypeIdFromEventTypes(EventTypeResult)
```

Get next available horse racing market and runner information using listMarketCatalogue

Common

```
Function GetListMarketCatalogueRequestString(ByVal EventTypeId As String) As String
    Dim dateNow As Date: dateNow = Format(Now, "yyyy-mm-dd hh:mm:ss")
    GetListMarketCatalogueRequestString = "{""filter"":{""eventTypeIds"":[""" & EventTypeId & """],""
marketCountries"":[""GB""],""marketTypeCodes"":[""WIN""]},""marketStartTime"":{""from"":""" & dateNow & """},""
sort"":""FIRST_TO_START"",""maxResults"":""1"",""marketProjection"":[""RUNNER_DESCRIPTION""]}"
End Function

Function GetMarketIdFromMarketCatalogue(ByVal Response As Object) As String
    GetMarketIdFromMarketCatalogue = Response.Item(1).Item("marketId")
End Function
```

JSON-RPC

```
Dim Request: Request = MakeJsonRpcRequestString(ListMarketCatalogueMethod, GetListMarketCatalogueRequestString
(EventTypeId))
Dim ListMarketCatalogueResponse As String: ListMarketCatalogueResponse = SendRequest(GetJsonRpcUrl(),
GetAppKey(), "", Request)
Dim MarketCatalogue: Set MarketCatalogue = ParseJsonRpcResponseToCollection(ListMarketCatalogueResponse)
Dim MarketId: MarketId = GetMarketIdFromMarketCatalogue(MarketCatalogue)
```

RESCRIPT

```
Public Const ListMarketCatalogueMethod As String = "listMarketCatalogue"
Dim Request: Request = GetListMarketCatalogueRequestString(EventTypeId)
Dim ListMarketCatalogueResponse As String: ListMarketCatalogueResponse = SendRequest(GetRestUrl() +
ListMarketCatalogueMethod, GetAppKey(), "", Request)
Dim MarketCatalogue: Set MarketCatalogue = ParseRestResponseToCollection(ListMarketCatalogueResponse)
Dim MarketId: MarketId = GetMarketIdFromMarketCatalogue(MarketCatalogue)
```

Get available back prices for the next horse racing Market using listMarketBook

Common

```
Function GetListMarketBookRequestString(ByVal MarketId As String) As String
    GetListMarketBookRequestString = "{""marketIds"":[""" & MarketId & """],""priceProjection"":{""priceData"":
[""EX_BEST_OFFERS""]}}"
End Function

Function GetSelectionIdFromMarketBook(ByVal Response As Object) As String
```

```
    Dim Runners As Object: Set Runners = Response.Item(1).Item("runners")
    GetSelectionIdFromMarketBook = Runners.Item(1).Item("selectionId")
    Set Runners = Nothing
End Function

Function GetAvailableToBackForSelection(ByVal SelectionId As String, ByVal Response As Object) As Collection
    Dim Runners As Object: Set Runners = Response.Item(1).Item("runners")

    Dim Index As Integer
    For Index = 1 To Runners.Count Step 1
        Dim Id: Id = Runners.Item(Index).Item("selectionId")
        If Id = SelectionId Then
            Set GetAvailableToBackForSelection = Runners.Item(Index).Item("ex").Item("availableToBack")
            Exit For
        End If
    Next

    Set Runners = Nothing
End Function
```

JSON-RPC

```
Dim Request: Request = MakeJsonRpcRequestString(ListMarketBookMethod, GetListMarketBookRequestString(MarketId))
Dim ListMarketBookResponse As String: ListMarketBookResponse = SendRequest(GetJsonRpcUrl(), GetAppKey(), "",
Request)

Dim MarketBook: Set MarketBook = ParseJsonRpcResponseToCollection(ListMarketBookResponse)
Dim SelectionId: SelectionId = GetSelectionIdFromMarketBook(MarketBook)

Dim AvailableToBack As Object: Set AvailableToBack = GetAvailableToBackForSelection(SelectionId, MarketBook)
```

RESCRIPT

```
Public Const ListMarketBookMethod As String = "listMarketBook"
Dim Request: Request = GetListMarketBookRequestString(MarketId)
Dim ListMarketBookResponse As String: ListMarketBookResponse = SendRequest(GetRestUrl() + ListMarketBookMethod,
GetAppKey(), "", Request)
Dim MarketBook: Set MarketBook = ParseRestResponseToCollection(ListMarketBookResponse)
Dim SelectionId: SelectionId = GetSelectionIdFromMarketBook(MarketBook)
Dim AvailableToBack As Object: Set AvailableToBack = GetAvailableToBackForSelection(SelectionId, MarketBook)
```

Place a bet on first runner from next horse racing market using placeOrders

Common

```
Function GetPlaceOrdersRequestString(ByVal MarketId As String, ByVal SelectionId As String, ByVal Price As
String) As String
    GetPlaceOrdersRequestString = "{""marketId"":""" & MarketId & """,""instructions"":[{""selectionId"":""" &
SelectionId & """,""handicap"":""0"",""side"":""BACK"",""orderType"":""LIMIT"",""limitOrder"":{""size"":""
0.01"",""price"":""" & Price & """,""persistenceType"":""LAPSE""}}]}"
End Function
```

JSON-RPC

```
Dim Price: Price = AvailableToBack.Item(1).Item("price")

Dim Request: Request = MakeJsonRpcRequestString(PlaceOrdersMethod, GetPlaceOrdersRequestString(MarketId,
SelectionId, Dim PlaceOrdersResponse As String: PlaceOrdersResponse = SendRequest(GetJsonRpcUrl(), GetAppKey(),
GetSession(), Request)

Dim PlaceExecutionReport: Set PlaceExecutionReport = ParseJsonRpcResponseToCollection(PlaceOrdersResponse)
Dim BetPlacementResult: BetPlacementResult = PlaceExecutionReport.Item("status")
```

RESCRIPT

```
Public Const PlaceOrdersMethod As String = "placeOrders"

Dim Price: Price = AvailableToBack.Item(1).Item("price")

Dim Request: Request = GetPlaceOrdersRequestString(MarketId, SelectionId, Price)
Dim PlaceOrdersResponse As String: PlaceOrdersResponse = SendRequest(GetRestUrl() + PlaceOrdersMethod,
GetAppKey(), GetSession(), Request)

Dim PlaceExecutionReport: Set PlaceExecutionReport = ParseRestResponseToCollection(PlaceOrdersResponse)
Dim BetPlacementResult: BetPlacementResult = PlaceExecutionReport.Item("status")
```

Other Common Code

```
Function ParseJsonRpcResponseToCollection(ByVal Response As String) As Object
    On Error GoTo ErrorHandler:
    Dim Lib As New JsonLib
    Set ParseJsonRpcResponseToCollection = Lib.parse(Response).Item("result")
    Exit Function

ErrorHandler:
    HandleError
End Function

Function ParseRestResponseToCollection(ByVal Response As String) As Object
    On Error GoTo ErrorHandler:
    Dim Lib As New JsonLib
    Set ParseRestResponseToCollection = Lib.parse(Response)
    Exit Function

ErrorHandler:
    HandleError
End Function

Sub HandleError()

    If Err.Number <> 0 Then
        AppendToLogFile "Error occurred: " & Err.Number & " - " & Err.Description
    End If

    End ' Exit the macro entirely

End Sub

Function MakeJsonRpcRequestString(ByVal Method As String, ByVal RequestString As String) As String
    MakeJsonRpcRequestString = "{""jsonrpc"": ""2.0"", ""method"": ""SportsAPING/v1.0/" & Method & """,
""params"": " & RequestString & ", ""id"": 1}"
End Function
```

Java

Java and API-NG

This page contains some code snippets of Java interaction with API-NG. This example shows how to use Java to send requests to list the event types, find the next horse racing market and then placing a bet with an invalid stake to trigger an error. The code referred to here is available at https://github.com/betfair/API-NG-sample-code/tree/master/java.

In the sample code we use the json-rpc and rescript protocols. All requests are sent and received using json format. To post a request we prepare the json string using Java objects and then we serialize the object using the gson library. The example requests contain some predefined data that can be modified depends on user needs.

Prerequisites

This is a maven project containing Java sample code to connect to the Betfair API-NG application.
It requires:
 - Apache Maven 3
 - Java 7

How to run it

You first build the project with:
mvn clean package

You can again use Apache Maven to run the application passing the app key, session token and the method (json-rpc or rescript):
mvn exec:java -Dexec.mainClass="com.betfair.aping.ApiNGDemo" -Dexec.args="<YOUR APP KEY> <YOUR SESSION TOKEN> <METHOD>"

<YOUR APP KEY>: a valid app key
<YOUR SESSION TOKEN>: a valid Betfair session token
<METHOD>: *json-rpc* or *rescript*

example: *mvn exec:java -Dexec.mainClass="com.betfair.aping.ApiNGDemo" -Dexec.args="myAppKey mySessionToken json-rpc"*

An addition code snippets for **Error Handling - non HTTP 200 responses** is included below, but does not form part of the code included the the git repository.

Code Snippet

Creating the request - JSON-RPC

```
String requestString;
//Handling the JSON-RPC request
JsonrpcRequest request = new JsonrpcRequest();
request.setId("1");
request.setMethod(ApiNGDemo.getProp().getProperty("SPORTS_APING_V1_0") + operation);
request.setParams(params);

requestString =  JsonConverter.convertToJson(request);
if(ApiNGDemo.isDebug())
         System.out.println("\nRequest: "+requestString);

//We need to pass the "sendPostRequest" method a string in util format:  requestString
HttpUtil requester = new HttpUtil();
return requester.sendPostRequestJsonRpc(requestString, operation, appKey, ssoToken);
```

Creating the request - Rescript

```
String requestString;
        //Handling the Rescript request
        params.put("id", 1);

        requestString =  JsonConverter.convertToJson(params);
        if(ApiNGDemo.isDebug())
            System.out.println("\nRequest: "+requestString);

        //We need to pass the "sendPostRequest" method a string in util format:  requestString
        HttpUtil requester = new HttpUtil();
        String response = requester.sendPostRequestRescript(requestString, operation, appKey, ssoToken);
        if(response != null)
            return response;
        else
            throw new APINGException();
```

Calling API-NG

```
String jsonRequest = param;
HttpPost post = new HttpPost(URL);
String resp = null;
try {
            post.setHeader(HTTP_HEADER_CONTENT_TYPE, ApiNGDemo.getProp().getProperty("APPLICATION_JSON"));
            post.setHeader(HTTP_HEADER_ACCEPT, ApiNGDemo.getProp().getProperty("APPLICATION_JSON"));
            post.setHeader(HTTP_HEADER_ACCEPT_CHARSET, ApiNGDemo.getProp().getProperty("ENCODING_UTF8"));
            post.setHeader(HTTP_HEADER_X_APPLICATION, appKey);
            post.setHeader(HTTP_HEADER_X_AUTHENTICATION, ssoToken);

            post.setEntity(new StringEntity(jsonRequest, ApiNGDemo.getProp().getProperty("ENCODING_UTF8")));

            HttpClient httpClient = new DefaultHttpClient();

            HttpParams httpParams = httpClient.getParams();
            HttpConnectionParams.setConnectionTimeout(httpParams, new Integer(ApiNGDemo.getProp().getProperty
("TIMEOUT")).intValue());
            HttpConnectionParams.setSoTimeout(httpParams, new Integer(ApiNGDemo.getProp().getProperty
("TIMEOUT")).intValue());

            resp = httpClient.execute(post, reqHandler);

        } catch (UnsupportedEncodingException e1) {
            //Do something

        } catch (ClientProtocolException e) {
            //Do something

        } catch (IOException ioE){
            //Do something

        }

        return resp;
```

Find Horse Racing event type id

```
MarketFilter marketFilter;
            marketFilter = new MarketFilter();
            Set<String> eventTypeIds = new HashSet<String>();

            System.out.println("1.(listEventTypes) Get all Event Types...\n");
            List<EventTypeResult> r = jsonOperations.listEventTypes(marketFilter, applicationKey, sessionToken);
            System.out.println("2. Extract Event Type Id for Horse Racing...\n");
            for (EventTypeResult eventTypeResult : r) {
                if(eventTypeResult.getEventType().getName().equals("Horse Racing")){
                    System.out.println("3. EventTypeId for \"Horse Racing\" is: " + eventTypeResult.
getEventType().getId()+"\n");
                    eventTypeIds.add(eventTypeResult.getEventType().getId().toString());
                }
            }
```

Get next available horse races:

```
System.out.println("4.(listMarketCataloque) Get next horse racing market in the UK...\n");
            TimeRange time = new TimeRange();
            time.setFrom(new Date());

            Set<String> countries = new HashSet<String>();
            countries.add("GB");

            Set<String> typesCode = new HashSet<String>();
            typesCode.add("WIN");

            marketFilter = new MarketFilter();
            marketFilter.setEventTypeIds(eventTypeIds);
            marketFilter.setMarketStartTime(time);
            marketFilter.setMarketCountries(countries);
            marketFilter.setMarketTypeCodes(typesCode);

            Set<MarketProjection> marketProjection = new HashSet<MarketProjection>();
            marketProjection.add(MarketProjection.COMPETITION);
            marketProjection.add(MarketProjection.EVENT);
            marketProjection.add(MarketProjection.EVENT_TYPE);
            marketProjection.add(MarketProjection.MARKET_DESCRIPTION);
            marketProjection.add(MarketProjection.RUNNER_DESCRIPTION);

            String maxResult = "1";

            List<MarketCatalogue> marketCatalogueResult = jsonOperations.listMarketCatalogue(marketFilter,
marketProjection, MarketSort.FIRST_TO_START, maxResult,
                    applicationKey, sessionToken);
```

Get list of runners in the market:

```
System.out.println("5. Print static marketId, name and runners....\n");
            printMarketCatalogue(marketCatalogueResult.get(0));
            /**
             * ListMarketBook: get list of runners in the market, parameters:
             * marketId:  the market we want to list runners
             *
             */
            System.out.println("6.(listMarketBook) Get volatile info for Market including best 3 exchange
prices available...\n");
            String marketIdChosen = marketCatalogueResult.get(0).getMarketId();

            PriceProjection priceProjection = new PriceProjection();
            Set<PriceData> priceData = new HashSet<PriceData>();
            priceData.add(PriceData.EX_ALL_OFFERS);
            priceData.add(PriceData.EX_BEST_OFFERS);
            priceData.add(PriceData.EX_TRADED);
            priceData.add(PriceData.SP_AVAILABLE);
            priceData.add(PriceData.SP_TRADED);

            //In this case we don't need these objects so they are declared null
            OrderProjection orderProjection = null;
            MatchProjection matchProjection = null;
            String currencyCode = null;

            List<String> marketIds = new ArrayList<String>();
            marketIds.add(marketIdChosen);
            List<MarketBook> marketBookReturn = jsonOperations.listMarketBook(marketIds, priceProjection,
                    orderProjection, matchProjection, currencyCode, applicationKey, sessionToken);
```

Place a bet:

```
long selectionId = 0;
            if ( marketBookReturn.size() != 0 ) {
                Runner runner = marketBookReturn.get(0).getRunners().get(0);
                selectionId = runner.getSelectionId();
                System.out.println("7. Place a bet below minimum stake to prevent the bet actually " +
                        "being placed for marketId: "+marketIdChosen+" with selectionId: "+selectionId+"...
\n\n");
                List<PlaceInstruction> instructions = new ArrayList<PlaceInstruction>();
                PlaceInstruction instruction = new PlaceInstruction();
                instruction.setHandicap(0);
                instruction.setSide(Side.BACK);
                instruction.setOrderType(OrderType.LIMIT);

                LimitOrder limitOrder = new LimitOrder();
                limitOrder.setPersistenceType(PersistenceType.LAPSE);
                //API-NG will return an error with the default size=0.01. This is an expected behaviour.
                //You can adjust the size and price value in the "apingdemo.properties" file
                limitOrder.setPrice(getPrice());
                limitOrder.setSize(getSize());

                instruction.setLimitOrder(limitOrder);
                instruction.setSelectionId(selectionId);
                instructions.add(instruction);

                String customerRef = "1";

                PlaceExecutionReport placeBetResult = jsonOperations.placeOrders(marketIdChosen, instructions,
customerRef, applicationKey, sessionToken);

                // Handling the operation result
                if (placeBetResult.getStatus() == ExecutionReportStatus.SUCCESS) {
                    System.out.println("Your bet has been placed!!");
                    System.out.println(placeBetResult.getInstructionReports());
                } else if (placeBetResult.getStatus() == ExecutionReportStatus.FAILURE) {
                    System.out.println("Your bet has NOT been placed :*( ");
                    System.out.println("The error is: " + placeBetResult.getErrorCode() + ": " + placeBetResult.
getErrorCode().getMessage());
                    System.out.println("Sorry, more luck next time\n\n");
                }
            } else {
                System.out.println("Sorry, no runners found\n\n");
            }
```

Error Handling - non HTTP 200 responses

The below guidance is for customers who are looking to handle additional HTTP responses other than the HTTP 200 success code when using Rescript requests.

1. In *RescriptResponseHandler* we need to modify the handleResponse method so that an HttpResponseException is thrown for unsuccessful response

```
public class RescriptResponseHandler implements
ResponseHandler<String> {


private static final String ENCODING_UTF_8 = "UTF-8";
```

```
    public String handleResponse(HttpResponse
response) throws ClientProtocolException, IOException {


        StatusLine statusLine =
response.getStatusLine();


        HttpEntity entity =
response.getEntity();


        String responseString =
EntityUtils.toString(entity,ENCODING_UTF_8);


        if
(statusLine.getStatusCode() != 200 ) {


           throw
new HttpResponseException(statusLine.getStatusCode(), responseString);


        }

        return entity == null ?
null : responseString;


    }

}
```

2. In *HttpUtil. sendPostRequest()* we catch the HttpResponseException, convert the Json response and eventually throw an APINGException containing the error details:

```
private String sendPostRequest(String param, String operation,
String appKey, String ssoToken, String URL, ResponseHandler<String>
reqHandler) throws APINGException {


        String jsonRequest =
param;

..….


……

resp = httpClient.execute(post, reqHandler);


        } catch
(UnsupportedEncodingException e1) {


//Do something



        }

catch(HttpResponseException ex){
```

```
ResponseError container = JsonConverter.convertFromJson(ex.getMessage(), new
TypeToken<ResponseError>() {}.getType());


APINGException  apingException=container.getDetail().getAPINGException();


if(apingException==null){

apingException= new APINGException();

apingException.setErrorCode(container.getFaultcode());

apingException.setErrorDetails(container.getFaultstring());


}

throw apingException;


         }


        catch
(ClientProtocolException e) {

//Do something



        } catch (IOException
ioE){


//Do something


        }

        return resp;


    }
```

3. *Notice that the error handling code from sendPostRequest uses a new class:* **ResponseError, which we are using to deserialize the error response**

```
public class ResponseError {


    private Detail detail;

    private String  faultcode;

    private String  faultstring;

    public String getFaultstring() {


        return faultstring;


    }


    public void setFaultstring(String faultstring)
{
        this.faultstring =
faultstring;


    }

    public String getFaultcode() {


        return faultcode;

    }

    public void setFaultcode(String faultcode) {


        this.faultcode =
faultcode;


    }


    public Detail getDetail() {

        return detail;


    }

    public void setDetail(Detail detail) {

        this.detail = detail;

    }


}


public class Detail {


    private APINGException APINGException;

    public APINGException getAPINGException() {

        return APINGException;

    }

    public void setAPINGException(APINGException
aPINGException) {


        APINGException =
aPINGException;

    }
```

Javascript

Pre requisites:

- nodejs - can be found at http://nodejs.org
- This document refers to the code found at https://github.com/betfair/API-NG-sample-code/tree/master/javascript.

JavaScript and API-NG

This page contains some code snippets of JavaScript interaction with API-NG. The example shows how to use JavaScript to construct and send requests to get list of markets, followed by list of runners for thechosen market and an attempt place a bet.

In the sample code we use the json-rpc and rescript protocols. All requests are sent and received using the JSON format. JavaScript uses node.js platform which allows us to run javascript from the command line. For more information see node.js home page at: http://nodejs.org/.

The application accepts session token and app key as command line arguments and runs in any linux environment that has nodejs installed.

How to run

In order to run this example, install nodejs which can be downloaded from here, set your environment path variable to point to the node executable and invoke a command:

```
/<path-to-your-nodejs-bin-directory>/node JsonRpcApiNgClient.js <your app key> <your session token>
```

Code Snippets

In order to construct the JSON-RPC request to API-NG we need to define mandatory headers that we send per request :

- appkey - your app key

- ssid - your session token

```
var options = {
    hostname: 'beta-api.betfair.com',
    port: 443,
    path: '/json-rpc',
    method: 'POST',
    headers: {
        'X-Application' : appkey,
        'Accept': 'application/json',
        'Content-type' : 'application/json',
        'X-Authentication' : ssid
        }
    }
```

Example POST request to get horse race id :

- jsonRequest - your constructed request object
- DEFAULT_ENCODING - set to "utf-8"

```
var requestFilters = '{"filter":{}}';
var jsonRequest = constructJsonRpcRequest('listEventTypes', requestFilters );
var req = https.request(options,function (res){
            res.setEncoding(DEFAULT_ENCODING);
            res.on('data', function (chunk) {
                str += chunk;
            });
            res.on('end', function (chunk) {
                // On resposne parse Json and check for errors
                var response = JSON.parse(str);
                handleError(response);
                // Retrieve id from response and get next available horse race
                getNextAvailableHorseRace(options, response);
            });

        });
        // Send Json request object
        req.write(jsonRequest, DEFAULT_ENCODING);
        req.end();

        req.on('error', function(e) {
            console.log('Problem with request: ' + e.message);
            return;
        });
    }
```

Example POST request to get next available horse races :

- eventId - retrieved form previous request
- jsonDate - is a current date

```
var requestFilters = '{"filter":{"eventTypeIds": [' + eventId + '],"marketCountries":["GB"],"marketTypeCodes":
["WIN"],"marketStartTime":{"from":"'+jsonDate+'"}},"sort":"FIRST_TO_START","maxResults":"1","marketProjection":
["RUNNER_DESCRIPTION"]}}';
var jsonRequest = constructJsonRpcRequest('listMarketCatalogue', requestFilters );
var req = https.request(options,function (res){
            res.setEncoding('utf8');
            res.on('data', function (chunk) {
                str += chunk;
            });

            res.on('end', function (chunk) {
                var response = JSON.parse(str);
                handleError(response);
                // Get list of runners that are available in that race
                getListOfRunners(options, response);
            });
        });
        req.write(jsonRequest, 'utf-8');
        req.end();
        req.on('error', function(e) {
            console.log('Problem with request: ' + e.message);
            return;
        });
```

Example POST request to get list of runners in the market :

- marketId - specify a market that we want to get runners from

```
var requestFilters = '{"marketIds":["' + marketId + '"],"priceProjection":{"priceData":["EX_BEST_OFFERS"],"
exBestOfferOverRides":{"bestPricesDepth":2,"rollupModel":"STAKE","rollupLimit":20},"virtualise":false,"
rolloverStakes":false},"orderProjection":"ALL","matchProjection":"ROLLED_UP_BY_PRICE"}}';
var jsonRequest = constructJsonRpcRequest('listMarketBook', requestFilters );
var str = '';
var req = https.request(options,function (res){
            res.setEncoding(DEFAULT_ENCODING);
            res.on('data', function (chunk) {
                str += chunk;
            });

            res.on('end', function (chunk) {
                var response = JSON.parse(str);
                handleError(response);
                // Place bet on first runner
                placeBet(options, response, marketId);
            });
```

```
        });
        req.write(jsonRequest, DEFAULT_ENCODING);
        req.end();
        req.on('error', function(e) {
            console.log('Problem with request: ' + e.message);
            return;
        });
```

Example POST request to place a bet on random runner :

- selectionId - runner that we want to place a bet on
- customerRef - unique reference for that transaction
- size - the size of the bet
- price - the price of the bet

```
var requestFilters = '{"marketId":"'+ marketId+'","instructions":[{"selectionId":"' + selectionId + '","
handicap":"0","side":"BACK","orderType":"LIMIT","limitOrder":{"size":"' + size + '","price":"' + price + '","
persistenceType":"LAPSE"}}],"customerRef":"'+customerRef+'"}}';
var jsonRequest = constructJsonRpcRequest('placeOrders', requestFilters );
var req = https.request(options,function (res){
            res.setEncoding(DEFAULT_ENCODING);
            res.on('data', function (chunk) {
                str += chunk;
            });
            res.on('end', function (chunk) {
                var response = JSON.parse(str);
                handleError(response);
                console.log(JSON.stringify(response, null, DEFAULT_JSON_FORMAT));
            });
        });
        req.write(jsonRequest, DEFAULT_ENCODING);
        req.end();
        req.on('error', function(e) {
            console.log('Problem with request: ' + e.message);
            return;
        });
```

- Pre requisites:
- JavaScript and API-NG
- How to run
- Code Snippets

PHP

Overview

The sample code is intended to demonstrate how you can utilise PHP to call the operations within API-NG and extract the desired output, it is very much a cut down sample and is not intended to be used in a production environment.

The code follows a simple workflow of finding the next horse racing market, displaying prices for the runners and then placing a bet with an invalid stake to trigger an error.

This documentation refers to the code available at https://github.com/betfair/API-NG-sample-code/tree/master/php.

Prerequisites

To run the sample code from the command line you must have a php5 cli installed along with the curl module enabled.

Debian linux Installation

In a Debian linux distro you can use the following commands to install the pre-requisites:

```
sudo apt-get update
sudo apt-get install php5-cli
sudo apt-get install php5-curl
```

Run the scripts

JSON-RPC

```
php -f jsonrpc.php <appkey> <sessiontoken>
```

Rescript

```
php -f rescript.php <appkey> <sessiontoken>
```

Code Snippets

Dealing with SSL in PHP

If you have errors relating to SSL certificate issues then you must do one of the following:

1) Quick fix for testing applications, should not be used in production as it may leave you exposed to man in the middle type attacks:

Add the following two lines to the sportsApingRequest function after the curl_init:

```
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
```

2) Correct fix for production applications:

You will need to make use of the CURLOPT_CAINFO option, and point it to the Betfair PEM formatted certificate (which you can export from your browser). The details of exporting the cert and using this option are beyond the scope of this document but can be found elsewhere online.

Calling API-NG with JSON-RPC protocol

Method and params values need to be change based on the required service operation. You can call batch multiple service operations together and correlate the responses with value of the id field.

```
function sportsApingRequest($appKey, $sessionToken, $operation, $params)
{
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, "https://api.betfair.com/exchange/betting/json-rpc/v1");
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array('Expect:',
        'X-Application: ' . $appKey,
        'X-Authentication: ' . $sessionToken,
        'Accept: application/json',
        'Content-Type: application/json'
    ));

    $postData =
        '[{ "jsonrpc": "2.0", "method": "SportsAPING/v1.0/' . $operation . '", "params" :' . $params . ', "id":
1}]';


    curl_setopt($ch, CURLOPT_POSTFIELDS, $postData);

    $response = json_decode(curl_exec($ch));
    curl_close($ch);

    if (isset($response[0]->error)) {
        echo 'Call to api-ng failed: ' . "\n";
        echo  'Response: ' . json_encode($response);
        exit(-1);
    } else {
        return $response;
    }

}
```

Calling API-NG with Rescript protocol

```
function sportsApingRequest($appKey, $sessionToken, $operation, $params)
{
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, "https://api.betfair.com/rest/v1/$operation/");
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array('Expect:',
        'X-Application: ' . $appKey,
        'X-Authentication: ' . $sessionToken,
        'Accept: application/json',
        'Content-Type: application/json'
    ));

    curl_setopt($ch, CURLOPT_POSTFIELDS, $params);

    $response = json_decode(curl_exec($ch));

    $http_status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
    curl_close($ch);

    if ($http_status == 200) {
        return $response;
    } else {
        echo 'Call to api-ng failed: ' . "\n";
        echo  'Response: ' . json_encode($response);
        exit(-1);
    }
}
```

Calling listEventTypes to obtain and extract Horse Racing Event Type ID - JSON-RPC

```
echo extractHorseRacingEventTypeId(getAllEventTypes($appKey, $sessionToken));

function getAllEventTypes($appKey, $sessionToken)
{

    $jsonResponse = sportsApingRequest($appKey, $sessionToken, 'listEventTypes', '{"filter":{}}');

    return $jsonResponse[0]->result;
}

function extractHorseRacingEventTypeId($allEventTypes)
{
    foreach ($allEventTypes as $eventType) {
        if ($eventType->eventType->name == 'Horse Racing') {
            return $eventType->eventType->id;
        }
    }
}
```

Calling listEventTypes to obtain and extract Horse Racing Event Type ID - Rescript

```
echo extractHorseRacingEventTypeId(getAllEventTypes($appKey, $sessionToken));

function getAllEventTypes($appKey, $sessionToken)
{

    $jsonResponse = sportsApingRequest($appKey, $sessionToken, 'listEventTypes', '{"filter":{}}');

    return $jsonResponse;
}

function extractHorseRacingEventTypeId($allEventTypes)
{
    foreach ($allEventTypes as $eventType) {
        if ($eventType->eventType->name == 'Horse Racing') {
            return $eventType->eventType->id;
        }
    }
}
```

Calling listMarketCatalogue to get next UK horse racing market and print the marketId and runners - JSON-RPC

```
printMarketIdAndRunners(getNextUkHorseRacingMarket($appKey, $sessionToken, $horseRacingEventTypeId);

function getNextUkHorseRacingMarket($appKey, $sessionToken, $horseRacingEventTypeId)
{

    $params = '{"filter":{"eventTypeIds":["' . $horseRacingEventTypeId . '"],
              "marketCountries":["GB"],
              "marketTypeCodes":["WIN"],
              "marketStartTime":{"from":"' . date('c') . '"}},
              "sort":"FIRST_TO_START",
              "maxResults":"1",
              "marketProjection":["RUNNER_DESCRIPTION"]}';

    $jsonResponse = sportsApingRequest($appKey, $sessionToken, 'listMarketCatalogue', $params);

    return $jsonResponse[0]->result[0];
}

function printMarketIdAndRunners($nextHorseRacingMarket)
{

    echo "MarketId: " . $nextHorseRacingMarket->marketId . "\n";
    echo "MarketName: " . $nextHorseRacingMarket->marketName . "\n\n";

    foreach ($nextHorseRacingMarket->runners as $runner) {
        echo "SelectionId: " . $runner->selectionId . " RunnerName: " . $runner->runnerName . "\n";
    }

}
```

Calling listMarketCatalogue to get next UK horse racing market and print the marketId and runners - Rescript

```
printMarketIdAndRunners(getNextUkHorseRacingMarket($appKey, $sessionToken, $horseRacingEventTypeId);

function getNextUkHorseRacingMarket($appKey, $sessionToken, $horseRacingEventTypeId)
{

    $params = '{"filter":{"eventTypeIds":["' . $horseRacingEventTypeId . '"],
              "marketCountries":["GB"],
              "marketTypeCodes":["WIN"],
              "marketStartTime":{"from":"' . date('c') . '"}},
              "sort":"FIRST_TO_START",
              "maxResults":"1",
              "marketProjection":["RUNNER_DESCRIPTION"]}';

    $jsonResponse = sportsApingRequest($appKey, $sessionToken, 'listMarketCatalogue', $params);

    return $jsonResponse[0];
}

function printMarketIdAndRunners($nextHorseRacingMarket)
{

    echo "MarketId: " . $nextHorseRacingMarket->marketId . "\n";
    echo "MarketName: " . $nextHorseRacingMarket->marketName . "\n\n";

    foreach ($nextHorseRacingMarket->runners as $runner) {
        echo "SelectionId: " . $runner->selectionId . " RunnerName: " . $runner->runnerName . "\n";
    }

}
```

Calling listMarketBook to get volatile price data and print the marketId and runners with best available prices - JSON-RPC

```
printMarketIdAndRunnersAndPrices($nextHorseRacingMarket, getMarketBook($appKey, $sessionToken, $marketId));

function getMarketBook($appKey, $sessionToken, $marketId)
{
    $params = '{"marketIds":["' . $marketId . '"], "priceProjection":{"priceData":["EX_BEST_OFFERS"]}}';

    $jsonResponse = sportsApingRequest($appKey, $sessionToken, 'listMarketBook', $params);

    return $jsonResponse[0]->result[0];
}

function printMarketIdRunnersAndPrices($nextHorseRacingMarket, $marketBook)
{

    function printAvailablePrices($selectionId, $marketBook)
    {

        // Get selection
        foreach ($marketBook->runners as $runner)
```

```
            if ($runner->selectionId == $selectionId) break;

        echo "\nAvailable to Back: \n";
        foreach ($runner->ex->availableToBack as $availableToBack)
            echo $availableToBack->size . "@" . $availableToBack->price . " | ";

        echo "\n\nAvailable to Lay: \n";
        foreach ($runner->ex->availableToLay as $availableToLay)
            echo $availableToLay->size . "@" . $availableToLay->price . " | ";

    }


    echo "MarketId: " . $nextHorseRacingMarket->marketId . "\n";
    echo "MarketName: " . $nextHorseRacingMarket->marketName;

    foreach ($nextHorseRacingMarket->runners as $runner) {
        echo "\n\n\n================================================================================\n";

        echo "SelectionId: " . $runner->selectionId . " RunnerName: " . $runner->runnerName . "\n";
        echo printAvailablePrices($runner->selectionId, $marketBook);
    }
}
```

Calling listMarketBook to get volatile price data and print the marketId and runners with best available prices - Rescript

```
printMarketIdAndRunnersAndPrices($nextHorseRacingMarket, getMarketBook($appKey, $sessionToken, $marketId));

function getMarketBook($appKey, $sessionToken, $marketId)
{
    $params = '{"marketIds":["' . $marketId . '"], "priceProjection":{"priceData":["EX_BEST_OFFERS"]}}';

    $jsonResponse = sportsApingRequest($appKey, $sessionToken, 'listMarketBook', $params);

    return $jsonResponse[0];
}

function printMarketIdRunnersAndPrices($nextHorseRacingMarket, $marketBook)
{

    function printAvailablePrices($selectionId, $marketBook)
    {

        // Get selection
        foreach ($marketBook->runners as $runner)
            if ($runner->selectionId == $selectionId) break;

        echo "\nAvailable to Back: \n";
        foreach ($runner->ex->availableToBack as $availableToBack)
            echo $availableToBack->size . "@" . $availableToBack->price . " | ";

        echo "\n\nAvailable to Lay: \n";
        foreach ($runner->ex->availableToLay as $availableToLay)
            echo $availableToLay->size . "@" . $availableToLay->price . " | ";

    }


    echo "MarketId: " . $nextHorseRacingMarket->marketId . "\n";
    echo "MarketName: " . $nextHorseRacingMarket->marketName;

    foreach ($nextHorseRacingMarket->runners as $runner) {
        echo "\n\n\n================================================================================\n";

        echo "SelectionId: " . $runner->selectionId . " RunnerName: " . $runner->runnerName . "\n";
        echo printAvailablePrices($runner->selectionId, $marketBook);
    }
}
```

Place bet on first runner of the market. Stake is below minimum to prevent actual bet placement - JSON-RPC

```
printBetResult(placeBet($appKey, $sessionToken, $marketId, $selectionId));

function placeBet($appKey, $sessionToken, $marketId, $selectionId)
{

    $params = '{"marketId":"' . $marketId . '",
                "instructions":
                     [{"selectionId":"' . $selectionId . '",
                       "handicap":"0",
                       "side":"BACK",
                       "orderType":
                       "LIMIT",
                       "limitOrder":{"size":"1",
                                    "price":"1000",
                                    "persistenceType":"LAPSE"}
                       }], "customerRef":"fsdf"}';

    $jsonResponse = sportsApingRequest($appKey, $sessionToken, 'placeOrders', $params);

    return $jsonResponse[0]->result;

}

function printBetResult($betResult)
{
    echo "Status: " . $betResult->status;

    if ($betResult->status == 'FAILURE') {
        echo "\nErrorCode: " . $betResult->errorCode;
        echo "\n\nInstruction Status: " . $betResult->instructionReports[0]->status;
        echo "\nInstruction ErrorCode: " . $betResult->instructionReports[0]->errorCode;
    } else
        echo "Warning!!! Bet placement succeeded !!!";
}
```

Place bet on first runner of the market. Stake is below minimum to prevent actual bet placement - Rescript

```
printBetResult(placeBet($appKey, $sessionToken, $marketId, $selectionId));

function placeBet($appKey, $sessionToken, $marketId, $selectionId)
{

    $params = '{"marketId":"' . $marketId . '",
                "instructions":
                     [{"selectionId":"' . $selectionId . '",
                       "handicap":"0",
                       "side":"BACK",
                       "orderType":
                       "LIMIT",
                       "limitOrder":{"size":"1",
                                    "price":"1000",
                                    "persistenceType":"LAPSE"}
                       }], "customerRef":"fsdf"}';

    $jsonResponse = sportsApingRequest($appKey, $sessionToken, 'placeOrders', $params);

    return $jsonResponse;

}

function printBetResult($betResult)
{
    echo "Status: " . $betResult->status;

    if ($betResult->status == 'FAILURE') {
        echo "\nErrorCode: " . $betResult->errorCode;
        echo "\n\nInstruction Status: " . $betResult->instructionReports[0]->status;
        echo "\nInstruction ErrorCode: " . $betResult->instructionReports[0]->errorCode;
    } else
        echo "Warning!!! Bet placement succeeded !!!";
}
```

Python

This documentation refers to the code available at https://github.com/betfair/API-NG-sample-code/tree/master/python.

Prerequisites:

1. python v 2.7.2 - http://www.python.org/getit/releases/2.7.2/
2. urllib2 python module - http://docs.python.org/2/library/urllib2.html
3. json python module - http://docs.python.org/2/library/json.html
4. datetime python module - http://docs.python.org/2/library/datetime.html
5. sys python module - http://docs.python.org/2/library/sys.html

A note on Python3:

We have added a python3 version of the json-rpc script, which is in the python subdirectory of the github sample code repo named ApiNgDemoJsonRpc-python3.py. This functions exactly the same way as the python 2.7X sample, but with compatibility tweaks for Python 3. The documentation below reflects the python 2.7X code, but the

Installation:

You only need to clone or download the repository linked to above. If you do not have a valid Python 2.7.X installation already then please follow the download and installation instructions from the python wiki.

Run the scripts

Change to the directory where you cloned the repository to and run the sample of your choice as follows:

JSON-RPC

```
python ApiNgDemoJsonRpc.py <appkey> <sessiontoken>
```

Rescript

```
python ApiNgDemoRescript.py <appkey> <sessiontoken>
```

Note: If the command line arguments for application key and session token are not provided then the script will prompt for application key and session token

Calling API-NG with JSON-RPC protocol

Method and param values need to be changed based on the required service operation.You can execute multiple service operation together with a single call using batch json-rpc call where you can correlate the responses with value of the id.

```
URL = url = "https://api.betfair.com/exchange/betting/json-rpc/v1"
jsonrpc_req = '{"jsonrpc": "2.0", "method": "SportsAPING/v1.0/listEventTypes", "params": {"filter":{ }}, "id": 1}'
headers = {'X-Application': appKey, 'X-Authentication': sessionToken, 'content-type': 'application/json'}

def callAping(jsonrpc_req):
    try:
        req = urllib2.Request(url, jsonrpc_req, headers)
        response = urllib2.urlopen(req)
        jsonResponse = response.read()
        return jsonResponse
    except urllib2.URLError:
        print 'Oops no service available at ' + str(url)
        exit()
    except urllib2.HTTPError:
        print 'Oops not a valid operation from the service ' + str(url)
        exit()
```

Calling API-NG with Rescript protocol

```
url = 'https://api.betfair.com/rest/v1.0/${operationName}/'
headers = {'X-Application': appKey, 'X-Authentication': sessionToken, 'content-type': 'application/json',
'accept': 'application/json'}
request = '{"filter":{"eventTypeIds":["7"],"marketCountries":["GB"],"marketStartTime":{"from":"2013-05-21T00:00:
00Z"}},"sort":"FIRST_TO_START","maxResults":"1","marketProjection":["RUNNER_METADATA"]}'

def callAping(url, request):
    try:
        req = urllib2.Request(url, request, headers)
        response = urllib2.urlopen(req)
        jsonResponse = response.read()
        return jsonResponse

    except urllib2.URLError:
        print 'Oops there is some issue with the request'
        exit()
    except urllib2.HTTPError:
        print 'Oops there is some issue with the request' + urllib2.HTTPError.getcode()
        exit()
```

Get next available horse racing market and runner information using listMarketCatalogue

JSON-RPC

```
def getMarketCatalogueForNextGBWin(eventTypeID):
    if (eventTypeID is not None):
        print 'Calling listMarketCatalouge Operation to get MarketID and selectionId'
        now = datetime.datetime.now().strftime('%Y-%m-%dT%H:%M:%SZ')
        market_catalogue_req = '{"jsonrpc": "2.0", "method": "SportsAPING/v1.0/listMarketCatalogue", "params":
{"filter":{"eventTypeIds":["' + eventTypeID + '"],"marketCountries":["GB"],"marketTypeCodes":["WIN"],'\

'"marketStartTime":{"from":"' + now + '"}},"sort":"FIRST_TO_START","maxResults":"1","marketProjection":
["RUNNER_METADATA"]}, "id": 1}'
        """
        print  market_catalogue_req
        """
        market_catalogue_response = callAping(market_catalogue_req)
        """
        print market_catalogue_response
        """
        market_catalouge_loads = json.loads(market_catalogue_response)
        try:
            market_catalouge_results = market_catalouge_loads['result']
            return market_catalouge_results
        except:
            print  'Exception from API-NG' + str(market_catalouge_results['error'])
            exit()


def getMarketId(marketCatalougeResult):
    if( marketCatalougeResult is not None):
        for market in marketCatalougeResult:
            return market['marketId']
```

```
def getSelectionId(marketCatalougeResult):
    if(marketCatalougeResult is not None):
        for market in marketCatalougeResult:
            return market['runners'][0]['selectionId']

marketCatalougeResult = getMarketCatalogueForNextGBWin(horseRacingEventTypeID)
marketid = getMarketId(marketCatalougeResult)
runnerId = getSelectionId(marketCatalougeResult)
```

Rescript

```
def getMarketCatalouge(eventTypeID):
    if(eventTypeID is not None):
        print 'Calling listMarketCatalouge Operation to get MarketID and selectionId'
        endPoint = 'https://api.betfair.com/rest/v1.0/listMarketCatalogue/'
        now = datetime.datetime.now().strftime('%Y-%m-%dT%H:%M:%SZ')
        market_catalouge_req = '{"filter":{"eventTypeIds":["' + eventTypeID + '"],"marketCountries":["GB"],"
marketStartTime":{"from":"' + now + '"}},"sort":"FIRST_TO_START","maxResults":"1","marketProjection":
["RUNNER_METADATA"]}'

        market_catalouge_response = callAping(endPoint, market_catalouge_req)

        market_catalouge_loads = json.loads(market_catalouge_response)
        return market_catalouge_loads


def getMarketId(marketCatalougeResult):
    if(marketCatalougeResult is not None):
        for market in marketCatalougeResult:
            return market['marketId']


def getSelectionId(marketCatalougeResult):
    if(marketCatalougeResult is not None):
        for market in marketCatalougeResult:
            return market['runners'][0]['selectionId']

marketCatalougeResult = getMarketCatalouge(horseRacingEventTypeID)
marketid = getMarketId(marketCatalougeResult)
runnerId = getSelectionId(marketCatalougeResult)
```

Get available price for the next horse racing market using listMarketBook

JSON-RPC

```
def getMarketBookBestOffers(marketId):
    print 'Calling listMarketBook to read prices for the Market with ID :' + marketId
    market_book_req = '{"jsonrpc": "2.0", "method": "SportsAPING/v1.0/listMarketBook", "params": {"marketIds":
["' + marketId + '"],"priceProjection":{"priceData":["EX_BEST_OFFERS"]}}, "id": 1}'
    """
    print  market_book_req
    """
    market_book_response = callAping(market_book_req)
    """
    print market_book_response
    """
    market_book_loads = json.loads(market_book_response)
    try:
        market_book_result = market_book_loads['result']
        return market_book_result
    except:
        print  'Exception from API-NG' + str(market_book_result['error'])
        exit()


def printPriceInfo(market_book_result):
    if(market_book_result is not None):
        print 'Please find Best three available prices for the runners'
        for marketBook in market_book_result:
            runners = marketBook['runners']
            for runner in runners:
                print 'Selection id is ' + str(runner['selectionId'])
                if (runner['status'] == 'ACTIVE'):
                    print 'Available to back price :' + str(runner['ex']['availableToBack'])
                    print 'Available to lay price :' + str(runner['ex']['availableToLay'])
                else:
                    print 'This runner is not active'

market_book_result = getMarketBookBestOffers(marketid)
printPriceInfo(market_book_result)
```

Rescript

```
def getMarketBook(marketId):
    if( marketId is not None):
        print 'Calling listMarketBook to read prices for the Market with ID :' + marketId
        market_book_req = '{"marketIds":["' + marketId + '"],"priceProjection":{"priceData":
["EX_BEST_OFFERS"]}}'

        endPoint = 'https://api.betfair.com/rest/v1.0/listMarketBook/'

        market_book_response = callAping(endPoint, market_book_req)

        market_book_loads = json.loads(market_book_response)
        return market_book_loads


def printPriceInfo(market_book_result):
    print 'Please find Best three available prices for the runners'
    for marketBook in market_book_result:
        try:
            runners = marketBook['runners']
            for runner in runners:
                print 'Selection id is ' + str(runner['selectionId'])
                if (runner['status'] == 'ACTIVE'):
                    print 'Available to back price :' + str(runner['ex']['availableToBack'])
                    print 'Available to lay price :' + str(runner['ex']['availableToLay'])
                else:
                    print 'This runner is not active'
        except:
            print 'No runners available for this market'

market_book_result = getMarketBook(marketid)
printPriceInfo(market_book_result)
```

Placing a bet on first active runner from next horse racing market using placeOrders

JSON-RPC

```
def placeFailingBet(marketId, selectionId):
    if( marketId is not None and selectionId is not None):
        print 'Calling placeOrder for marketId :' + marketId + ' with selection id :' + str(selectionId)
        place_order_Req = '{"jsonrpc": "2.0", "method": "SportsAPING/v1.0/placeOrders", "params":
{"marketId":"' + marketId + '","instructions":'\

'[{"selectionId":"' + str(
            selectionId) + '","handicap":"0","side":"BACK","orderType":"LIMIT","limitOrder":{"size":"0.01","
price":"1.50","persistenceType":"LAPSE"}}],"customerRef":"test12121212121"}, "id": 1}'
        """
        print place_order_Req
        """
        place_order_Response = callAping(place_order_Req)
        place_order_load = json.loads(place_order_Response)
        try:
            place_order_result = place_order_load['result']
            print 'Place order status is ' + place_order_result['status']
            """
            print 'Place order error status is ' + place_order_result['errorCode']
            """
            print 'Reason for Place order failure is ' + place_order_result['instructionReports'][0]
['errorCode']
        except:
            print  'Exception from API-NG' + str(place_order_result['error'])


placeBet(marketid, runnerId)
```

Rescript

```
def placeBet(marketId, selectionId):
    if( marketId is not None and selectionId is not None):
        print 'Calling placeOrder for marketId :' + marketId + ' with selection id :' + str(selectionId)
        place_order_Req = '{"marketId":"' + marketId + '","instructions":'\
                                                       '[{"selectionId":"' + str(
            selectionId) + '","handicap":"0","side":"BACK","orderType":"LIMIT","limitOrder":{"size":"1.01","
price":"1.50","persistenceType":"LAPSE"}}],"customerRef":"test12121212121"}'
        endPoint = 'https://api.betfair.com/rest/v1.0/placeOrders/'

        place_order_Response = callAping(endPoint, place_order_Req)
        place_order_load = json.loads(place_order_Response)
        print 'Place order status is ' + place_order_load['status']

        print 'Reason for Place order failure is ' + place_order_load['instructionReports'][0]['errorCode']


placeBet(marketid, runnerId)
```

Developer Support

- Support Resources
- Contact Information
- Information Required for Troubleshooting

Support Resources

Please check the following resource before contacting us:

API System Status - check the status of the Exchange API services.
Getting Started Guide - quick start guide for the Exchange API, including request & response examples.
FAQ's - search our knowledge base for the answer to your questions.
Developer Forum - discuss your issue with our experienced developer forum community.
Reference Guide - the latest documentation for the Exchange API.
Sample Code - available in a number of programming languages.
API Developer Tools - quickly test API operations via a simple interface.
Betfair Partnerships - to track activity and players to your Affiliate and Partnerships account, please contact your Account Manager who will supply you with the necessary links and formats. Please note: if you do not apply the correct links or parameters you will not receive affiliate commission or earnings from the players you have referr

Contact Information

If these resources don't help and you need further **API Support**, please contact us using the **Contact Developer Support** link below:

For general queries regarding your **Betfair account,** please contact the Betfair Customer Service team.

- **Betfair Customer Service - Betfair Help Centre**
- **Betfair Customer Service Twitter Available 08:00 - 23:00 (BST/GMT)**
- **Betfair Partnerships -** to track activity and players to your Affiliate and Partnerships account, please contact your Account Manager who will supply you with the necessary links and formats. Please note: if you do not apply the correct links or parameters you will not receive affiliate commission or earnings from the players you have referred.

- **Contact Developer Support - 09:00-17:00 (BST/GMT) Monday - Friday (excluding. UK bank holidays)**

Information Required for Troubleshooting

When raising API issues with **Developer Support** please provide the following details as this will help us to fully investigate any issues in a timely manner:

Your details:

- Applicaton Key used to make the request/s
- The ErrorCode/UUID (if exception) e.g. **"errorCode : UNEXPECTED_ERROR requestUUID : prdang007-11130238-0005ef4437**"

API Request details:

- Date/Time of the issue with timezone
- Operation/Endpoint URL
- JSON request
- JSON response
- Your Application logs (if applicable)