

API v6 Reference Guide

Contents

| | |
|---|-----------|
| Betfair API Introduction..... | 5 |
| Overview..... | 5 |
| WSDL Location..... | 5 |
| The Connection End-point URLs for the API Services..... | 5 |
| Betting on Australian Events..... | 6 |
| What's New in API 6..... | 7 |
| What's New in this Release..... | 9 |
| Where to Find More Information..... | 10 |
| Session Management..... | 10 |
| Betfair's Load Balancing of Client Requests..... | 10 |
| API Login Sessions..... | 10 |
| General API Services Reference..... | 13 |
| API Request/Response Header..... | 13 |
| Login (global)..... | 14 |
| Service Details..... | 14 |
| Logout (global)..... | 17 |
| Service Details..... | 17 |
| KeepAlive (global)..... | 18 |
| Service Details..... | 18 |
| Read-Only Betting API Services Reference..... | 20 |
| Convert Currency (global)..... | 20 |
| Service Details..... | 20 |
| Get Active Event Types (global)..... | 21 |
| Service Details..... | 21 |
| Get All Currencies (global)..... | 23 |
| Service Details..... | 23 |
| Get All Currencies V2 (global)..... | 24 |
| Service Details..... | 24 |
| Get All Event Types (global)..... | 25 |
| Service Details..... | 25 |
| Get All Markets (exchange)..... | 27 |
| Service Details..... | 27 |
| Get Bet (exchange)..... | 31 |
| Service Details..... | 31 |
| Get Bet History (exchange)..... | 35 |
| Service Details..... | 35 |
| Get Bet Lite (exchange)..... | 41 |
| Service Details..... | 41 |
| Get Bet Matches Lite (exchange)..... | 43 |
| Service Details..... | 43 |
| Get Complete Market Prices Compressed (exchange)..... | 44 |
| Service Details..... | 44 |
| Get Current Bets (exchange)..... | 49 |
| Service Details..... | 49 |
| Get Current Bets Lite (exchange)..... | 55 |
| Service Details..... | 55 |
| Get Detail Available Market Depth (exchange)..... | 58 |
| Service Details..... | 58 |
| Get Events (global)..... | 61 |
| Service Details..... | 62 |
| Get In-Play Markets (exchange)..... | 66 |

| | |
|---|------------|
| Service Details..... | 66 |
| Get Market (exchange)..... | 70 |
| Service Details..... | 70 |
| Get Market Info (exchange)..... | 74 |
| Service Details..... | 74 |
| Get Market Prices (exchange)..... | 76 |
| Service Details..... | 76 |
| Get Market Prices Compressed (exchange)..... | 80 |
| Service Details..... | 80 |
| Get Matched and Unmatched Bets (exchange)..... | 85 |
| Service Details..... | 85 |
| Get Market Profit And Loss (exchange)..... | 90 |
| Service Details..... | 91 |
| Get Market Traded Volume (exchange)..... | 94 |
| Service Details..... | 94 |
| Get Market Traded Volume Compressed (exchange)..... | 96 |
| Service Details..... | 96 |
| Get Private Markets (exchange)..... | 99 |
| Service Details..... | 99 |
| Get Silks (exchange)..... | 101 |
| Service Details..... | 101 |
| Get Silks V2 (exchange)..... | 104 |
| Service Details..... | 104 |
| Bet Placement API Services Reference..... | 108 |
| Cancel Bets (exchange)..... | 108 |
| Service Details..... | 108 |
| Cancel Bets By Market (exchange)..... | 111 |
| Service Details..... | 111 |
| Place Bets (exchange)..... | 113 |
| Bet Types..... | 113 |
| Bet Categories..... | 114 |
| Bet Persistence..... | 114 |
| Service Details..... | 114 |
| Update Bets (exchange)..... | 121 |
| Service Details..... | 122 |
| Account Management API Services Reference..... | 127 |
| Add Payment Card (global)..... | 127 |
| Service Details..... | 127 |
| Delete Payment Card (global)..... | 131 |
| Service Details..... | 131 |
| Deposit From Payment Card (global)..... | 133 |
| Service Details..... | 133 |
| Forgot Password (global)..... | 135 |
| Service Details..... | 136 |
| Get Account Funds (exchange)..... | 137 |
| Service Details..... | 137 |
| Get Account Statement (exchange)..... | 139 |
| Service Details..... | 139 |
| Get Payment Card (global)..... | 142 |
| Service Details..... | 142 |
| Get Subscription Info (global)..... | 145 |
| Service Details..... | 145 |
| Modify Password (global)..... | 146 |
| Service Details..... | 146 |
| Modify Profile (global)..... | 148 |
| Service Details..... | 148 |

| | |
|--|------------|
| Retrieve LIMB Message (global)..... | 151 |
| Service Details..... | 151 |
| Self Exclude (global)..... | 155 |
| Service Details..... | 155 |
| Set Chat Name(global)..... | 156 |
| Service Details..... | 156 |
| Submit LIMB Message (global)..... | 158 |
| Service Details..... | 158 |
| Transfer Funds (global)..... | 163 |
| Service Details..... | 163 |
| Update Payment Card (global)..... | 166 |
| Service Details..... | 166 |
| View Profile (global)..... | 169 |
| Service Details..... | 169 |
| View Profile V2(global)..... | 171 |
| Service Details..... | 172 |
| View Refer And Earn(global)..... | 175 |
| Service Details..... | 175 |
| Withdraw To Payment Card (global)..... | 176 |
| Service Details..... | 176 |
| Appendix..... | 179 |
| Additional Information..... | 179 |
| Time zones..... | 179 |
| Currency..... | 180 |
| Line and Range Markets..... | 180 |
| Timestamps..... | 180 |
| Horse Racing Non-Runners..... | 180 |
| Betting In-Running..... | 180 |
| Locale Specification..... | 181 |
| Exponential Numbers..... | 181 |
| Betfair Simple Data Types..... | 182 |
| AccountStatementEnum..... | 182 |
| AccountStatementIncludeEnum..... | 182 |
| AccountStatusEnum..... | 182 |
| AccountTypeEnum..... | 183 |
| BetCategoryTypeEnum..... | 183 |
| BetPersistenceTypeEnum..... | 184 |
| BetsOrderByEnum..... | 184 |
| BetStatusEnum..... | 184 |
| BetTypeEnum..... | 185 |
| BillingPeriodEnum..... | 185 |
| CardTypeEnum..... | 185 |
| GamcareLimitFreqEnum..... | 185 |
| GenderEnum..... | 186 |
| MarketStatusEnum..... | 186 |
| MarketTypeEnum..... | 186 |
| MarketTypeVariantEnum..... | 187 |
| PaymentCardStatusEnum..... | 187 |
| RegionEnum..... | 187 |
| SecurityQuestion1Enum..... | 187 |
| SecurityQuestion2Enum..... | 188 |
| ServiceEnum..... | 188 |
| SortOrderEnum..... | 189 |
| SubscriptionStatusEnum..... | 189 |
| TitleEnum..... | 189 |
| ValidationErrorsEnum..... | 190 |

Betfair API Introduction

Overview

This document outlines the services that are available via the Betfair Application Programmers Interface (API) Version 6. The API is a SOAP interface that is available over a secure web connection.

WSDL Location

A WSDL defines the interface for a SOAP Web services API. Normally, you use the WSDL file with a programming environment, such as Visual Studio .Net or Axis, that generates classes or objects for calling the API. The Betfair SOAP API is divided between two sets of services available on three separate endpoints: global and two exchange specific. As such, there are three Web Service Description Language (WSDL) files available at the following URLs:

- Global: <https://api.betfair.com/global/v3/BFGlobalService.wsdl>
- UK Exchange: <https://api.betfair.com/exchange/v5/BFExchangeService.wsdl>
- Aus. Exchange: <https://api-au.betfair.com/exchange/v5/BFExchangeService.wsdl>



Note:

The UK and Australia WSDL files define the same set of exchange-specific services, but they specify different endpoint URLs.

A separate WSDL has been provided for each exchange and the global services:

- The global services are used to log in and out, administer your Betfair account and funds, and to navigate to the sports events you want to bet on.
- The exchange services are those that enable you to place your bets as conveniently and quickly as possible. They include the services for viewing betting markets, placing, updating and cancelling bets, viewing your betting history, and checking your available funds and account statement.

All the services available from the API are described in the following pages of this reference guide. There is one chapter for each service, and the chapters each contain a brief description followed by a list of the input and output parameters for the service in question: the input parameters are the data you must include in the service request, and the output parameters are the data that the API returns in response to a service request. Each chapter indicates in brackets in its title whether the service the chapter is about is a global or an exchange service.

The Connection End-point URLs for the API Services

The Betfair sports betting API services are available only over SSL secure connections. For release 5.0 of the API, there are three connection end-point URLs:

- <https://api.betfair.com/global/v3/BFGlobalService>

This URL is for requesting the API's global services (see above)

- <https://api.betfair.com/exchange/v5/BFExchangeService>

This URL is for requesting the API's exchange services that enable you to bet on UK and other (but not Australian) sports events,

- <https://api-au.betfair.com/exchange/v5/BFExchangeService>

This URL is for requesting the API's exchange services that enable you to bet on Australian sports events (that is, on events that are taking place in Australia or that - for some other reason - have a betting market that operates under an Australian licence).

Betting on Australian Events

In API release 5.0 the API changed to support betting on events within Australia and/or under the jurisdiction of the Tasmanian Gaming Commission (TGC). This section explains these changes.

The Requirements of the Tasmanian Gaming Commission (TGC)

The main requirements of are that:

1. Betting markets for sporting events that take place in Australia must be operated under an Australian licence from the Tasmanian Gaming Commission (the body that regulates on-line gambling in Australia).
2. Bets placed in a betting market that operates under an Australian licence must be matched and settled on equipment residing physically in Australia.
3. No performance degradation may be suffered by customers (in their placement and management of bets) as a result of the requirement for bets to be matched and settled in Australia.
4. Providers of on-line gambling services must formally verify the name and address of every customer who wishes to bet on an Australian sporting event. Until this verification has taken place for a customer, that customer is not permitted to:
 - Deposit more than \$300 (USD) in a single month into his or her betting account
 - Use the funds in the betting account to place bets whose cumulative cost in a single month exceeds \$300 (USD)
 - Remove any funds from the account.
 - Finally, the name and address verification process must be completed within three months of the first deposit into the account, otherwise the account will be suspended until verification has taken place.

The Two Exchange Servers and Three Endpoints

To meet the first and second of the TGC's requirements, from now on you can only bet on Australian events by accessing the Australian exchange server. The API enables you to do this using the session token you received with the last valid API service response (whichever service of the API it was from), but you need to make sure that your client software sends its request to the correct endpoint URL for the Australian exchange server. You must also explicitly specify the correct endpoint url. This URL is as follows:

```
https://api-au.betfair.com/exchange/v5/BFExchangeService
```

Similarly, to bet on sporting events taking place in the UK or anywhere else in the world (except Australia), your client software must request exchange services from the UK exchange server. Again, the API enables you to use the last valid session token you received, but you must make sure that your client software sends its request to the correct endpoint URL for the UK exchange server. You must also explicitly specify the correct endpoint URL. This URL is as follows:

```
https://api.betfair.com/exchange/v5/BFExchangeService
```

It is simple to find out which exchange server is hosting a betting market. When you use the GetEvents service to navigate to an event you want to bet on, the API's response to your GetEvent request now includes a parameter called `exchangeId`. This parameter has the value 1 if the betting market is hosted by the UK exchange server, and the value 2 if it is hosted by the Australian exchange server.

When you have discovered which exchange server is hosting the betting market you want, you need to send a GetMarket service request to that exchange server. A consequence of the need to make the betting exchange services available from geographically separate exchange servers has been that all the non-betting-related API services (i.e., those services concerned with logging in and out of the API, administering your account, and navigating to particular sports events) are now also logically separate from the betting exchange services.

This logical separation of the API services into betting-related (exchange-specific) services and non-betting-related (non-exchange-specific and, in that sense, “global”) services has been formalized by the creation of a discrete domain for the API’s global services. This is why there is now a new endpoint URL for logging into the API and accessing the global services:

<https://api.betfair.com/global/v3/BFGlobalService>

For lists of the services that are included amongst the “global” services and “exchange” services, see the individual description of the service.

Finally, this distinction between the two types of services available is also now reflected in the provision of three distinct WSDL files for the API: there is one for the global services and two for each exchange service, although the exchange WSDLs only reflect the difference in the endpoints. All services between the Australian and UK exchanges are the same.

Using Region-specific Wallets for Placing Bets

The third requirement of the Tasmanian Gaming Commission (as listed above) is that customers must experience no performance degradation (in the placement and management of bets) as a consequence of their bets on Australian events being matched and settled on equipment that resides physically in Australia.

To meet this requirement, Betfair have introduced the concept of region-specific betting “wallets”.

A wallet is stored locally to an exchange server and contains all the user information the exchange server needs in order to execute a betting request. It contains, for example, the funds that the requesting user has available to bet with on the exchange server, and it also contains details of the loss and exposure limits that apply to the user’s bets on that exchange server. The Australian wallet exists to enable customers to bet on Australian events.



Note:

The only funds available to you for betting on a particular exchange server are the funds in the wallet that is local to that exchange server. If you have a high balance in your UK wallet but no funds in your Australian wallet, you cannot place a bet on an Australian event. Similarly, if you have a high balance in your Australian wallet but no funds in your UK wallet, you cannot place a bet on the Betfair UK exchange server (which is where the betting markets for all events - except Australian ones - are hosted).

A new service has been introduced in API 5 that enables you to transfer funds between the Australian and UK wallets. This is the TransferFunds service. The only way to put funds into your Australian wallet is to transfer them from your UK wallet into the Australian wallet by using this service. Also, the only way to withdraw funds from your Australian wallet is to transfer them from your Australian wallet into the UK wallet by using the TransferFunds service.

The Need for Account Holders to Confirm their Identity

Finally, the fourth requirement of the Tasmanian Gaming Commission was that all customers placing bets on Australian events must confirm their name and address within three months of making their first deposit into their Australian wallet.

To meet this requirement Betfair corresponds with all customers, requesting them to confirm the name and address that Betfair have for them. Once this process has been completed, the restrictions listed are lifted from a customer’s Australian wallet.

What’s New in API 6

This section describes the changes and enhancements from release 5 of the API.

Betfair Starting Price Betting

With API 6, Betfair has introduced a new type of bet, the Betfair Starting Price. Betfair have introduced the Betfair Starting Price (BSP) for selected markets that allows us to offer a starting price bet by reconciling odds for a selection at the time the event starts.

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.

Bet Persistence

Previously on Betfair, we would try to subsequently match all bets that were fully or partially unmatched, but these would 'lapse' and be cancelled at the start of the event.

You can still choose for this to happen if you wish but we are giving that decision to you to make. Alternatively, you can decide to make use of our Bet Persistence functionality and keep the bet once the market turns in play. This allows you more of an opportunity to get your bet on although there is still no guarantee you will get your price if the market goes against you.

You can also choose to convert your unmatched exchange bet to an SP bet at the start of the event which will ensure it goes on and gets matched.

New Service

A new API service getSilks is now available to API users who wish to publish UK and Irish horse racing data such as jockey colours in their applications. The following information is now available in the getSilks response:

- selectionId
- silksURL
- silksText
- trainerName
- ageWeight
- form
- daysSince
- jockeyClaim
- wearing
- saddlecloth
- stallDraw

These response fields will not change. However, we are planning to add new fields to the response in late February so please ensure your implementation does not break when these additional fields are returned in the getSilks response.

Changes to Existing Services

The following table shows the services that have changed between API 5.x and API 6.0.

Table 1: New Services and Key Changes to Existing Services

| API 6.0 Service | Description |
|---|--|
| <i>Get Market (exchange) and Get Market Info (exchange)</i> | A new field indicates if the market is enabled for BSP bets. |
| <i>Get Market Prices (exchange)</i> | Updated to return the Near and Far BSP, and the actual BSP (after the market turns in-play). |
| <i>Get Market Prices Compressed (exchange)</i> | Updated to return the Near and Far BSP, and the actual BSP (after the market turns in-play). |
| <i>Get Market Profit And Loss (exchange)</i> | Updated to allow for inclusion of BSP bets. |
| <i>Get Detail Available Market Depth (exchange)</i> | Updated to show the amount of money bet as BSP back and lay bets and, post-reconciliation, the amount of money matched at the starting price. |
| <i>Get Market Traded Volume (exchange)</i> | Updated to show the amount of money traded at the BSP when the market turns in-play. |
| <i>Place Bets (exchange)</i> | Updated to allow you to place BSP bets and exchange bets that persist in-play or convert to BSP bets. |
| <i>Update Bets (exchange)</i> | Updated to allow you to edit BSP Limit on Close bets or to change the persistence of an unmatched exchange bet. |
| <i>Cancel Bets (exchange)</i> | Updated to handle attempts to cancel BSP bets |
| <i>Get Bet (exchange) (and Lite version)</i> | Updated to return information on BSP bets, including liability, and the persistence of exchange bets. |
| <i>Get Matched and Unmatched Bets (exchange) (and Lite version)</i> | Updated to return information on BSP bets, including liability, and the persistence of exchange bets. Also includes the transactionId for each portion of a bet. |
| <i>Get Current Bets (exchange) (and Lite version)</i> | Updated to return information on BSP bets, including liability, and the persistence of exchange bets. |
| <i>Get Bet History (exchange)</i> | Updated to return information on BSP bets, including liability, and the persistence of exchange bets. |
| <i>Get Account Statement (exchange)</i> | Updated to return information on BSP bets, including liability. |

What's New in this Release

This release of the API includes the following:

New Services

This releases includes a number of new services that extend the functionality of existing API services. These services return the same data as the existing services, but include additional details. We have introduced these services to extend the API without breaking the API for existing applications.

These extended services are:

- [Get All Currencies V2 \(global\)](#)
- [Get Silks V2 \(exchange\)](#)
- [View Profile V2\(global\)](#)

In addition, this releases introduces a new service for internal Betfair applications to manage the Betfair user's forum name.

- [Set Chat Name\(global\)](#)

Where to Find More Information

There are a number of places you can look for more information on the Betfair API and web services in general. The following table lists additional documentation available from Betfair and some third-party organizations.

Table 2: More Information

| For information on: | See: |
|---------------------------------|---|
| Betfair API developer resources | http://bdp.betfair.com/apidocumentation.php |
| W3C SOAP Specification | http://www.w3.org/TR/soap/ |
| IETF SSL/TLS RFC | http://www.ietf.org/rfc/rfc3546.txt?number=3546 |

Session Management

This chapter gives an overview of the methods used by the API to maintain client login sessions. It includes some best-practice information you should follow to get the best performance for your applications.

Specifically, this chapter covers:

- How Betfair load balances client requests for the API's global and exchange services
- Session tokens

Betfair's Load Balancing of Client Requests

The Betfair API applications providing the global services and the exchange services are all run on application servers that are located behind a load balancer.

There is one load balancer for the global services, one for the UK exchange server, and one for the Australian exchange server. In each of these cases, the load balancer distributes service requests across the available application servers.

API Login Sessions

This section describes how the API handles sessions. It describes how to handle the `sessionToken` returned by the API, how to make sure sessions do not time out and how to handle multi-threaded applications with sessions.

Session Token Handling

After a successful log-in, the Betfair API Login service returns a session token in the `sessionToken` field. The session token is used to maintain state information. It acts as a key proving that a successful login has been performed. A session token is returned in the response to every request.

The `sessionToken` normally remains the same from one call to the next, however, the API may issue a new `sessionToken` in any response. If a new `sessionToken` is issued, your application must use the new `sessionToken` in subsequent responses.

Procedure for handling session tokens

You might handle session tokens in your application as follows:

1. Store the session token from the Login response
2. Send the session token in the request for the next API call
3. When the API responds, compare the returned session token with the one you received from the Login response
4. If the session token has not changed, continue using the same session token. If the session token has changed, you must now use the newly returned session token for subsequent calls.
5. If your application has not made any calls for about twenty minutes, call the API KeepAlive service to prevent your session from timing out.

The following flowchart shows how you might handle sessions in your application:



Note:

You should not rely on the sessionToken to remain static for the length of your application run.

A sessionToken returned by one of the API's global services (for example, GetEvents) is both valid and required for requesting one of the exchange services, even though these different services are now only available from different endpoint URLs. (For information about the distinction between global and exchange services see [Betting on Australian Events](#).)

Keeping Sessions Alive

The API maintains a session for a variable period of time, currently about an hour. After a period of no activity, the session expires. Whenever your application calls the API, the timer is reset. You can handle this situation in one of three ways:

- You can detect that a session has expired by looking for the NO_SESSION response in the API Response Header and, if the session has expired, call the Login service again to re-establish your session.
- You can make sure your application always makes an API call within twenty minutes of the previous call.
- If your application does not otherwise need to call the API within twenty minutes, you can use the API KeepAlive service to reset the session time out.

Ending a Session

Ending a session logs the user out of the exchange and invalidates the session token. When the session token is invalidated, subsequent calls to the API using the session token do not succeed.

You should explicitly call the API Logout service when your application shuts down. Also, it is recommended that your application provide a way for the user to log out on demand.

Example API Request and Response

The following example shows an example XML request and response for the login service. The request is:

```
<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <login xmlns="http://www.betfair.com/publicapi/v3/BFGlobalService/">

      <request>

        <locationId xmlns="">0</locationId>

        <password xmlns="">bfpasword</password>

        <productId xmlns="">82</productId>

        <username xmlns="">busername</username>
```

```

        <vendorSoftwareId xmlns="">0</vendorSoftwareId>

    </request>

</login>

</soap:Body>

</soap:Envelope>

```

The response received is:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:n2="http://www.betfair.com/publicapi/types/global/v3/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <n:loginResponse xmlns:n="http://www.betfair.com/publicapi/v3/BFGlobalService/">

      <n:Result xsi:type="n2:LoginResp">

        <header xsi:type="n2:APIResponseHeader">

          <errorCode xsi:type="n2:APIErrorEnum">OK</errorCode>

          <minorErrorCode xsi:nil="1">

          </minorErrorCode>

          <sessionToken xsi:type="xsd:string">PKdvAW+ruIAg0s769DeM+vxY=</
sessionToken>

          <timestamp xsi:type="xsd:dateTime">2007-08-23T11:25:05.285Z</
timestamp>

        </header>

        <currency xsi:type="xsd:string">GBP</currency>

        <errorCode xsi:type="n2:LoginErrorEnum">OK</errorCode>

        <minorErrorCode xsi:nil="1">

        </minorErrorCode>

        <validUntil xsi:type="xsd:dateTime">0001-01-01T00:00:00.000Z</
validUntil>

      </n:Result>

    </n:loginResponse>

  </soap:Body>

</soap:Envelope>

```

In this example, the `sessionToken` contains the value `PKdvAW+ruIAg0s769DeM+vxY=`.

General API Services Reference

API Request/Response Header

The API Request Header and the API Response Header contain the user's session token and client stamp which uniquely identify each call/session. There is an instance of `APIResponseHeader` returned in the output for each service call.



Note:

You must make sure that, in addition to the API Response Header `errorCode`, that you check for errors in the service specific `errorCode` as well.

The following table describes the `APIRequestHeader`.

Table 3: APIRequestHeader

| Parameter | Type | Description |
|--------------|--------|--|
| clientStamp | Long | Not currently used and should be set to zero |
| sessionToken | String | Session Identifier. For more information, see Session Management . |

The following table describes the `APIReponseHeader`

Table 4: APIResponseHeader

| Parameter | Type | Description |
|----------------|------------------------------|---|
| errorCode | APIErrorEnum | If not null, indicates a non service specific error has occurred. Make sure you also check the service specific <code>errorCode</code> field for any service specific errors. |
| minorErrorCode | String | Reserved for future use - currently always null |
| sessionToken | String | Uniquely identifier for next request in this session. A valid session token must be set in the <code>APIRequestHeader</code> for the next service invoked. For more information, see Session Management |
| timestamp | DateTime | This field is deprecated and it should not be relied on for determining the order or age of responses. |

The following table describes the possible error codes returned by the Betfair API in the response header. Make sure you check the response error code for the particular service as well.

Table 5: APIErrorEnum

| Value | Description |
|--|---|
| OK | |
| INTERNAL_ERROR | Internal error |
| EXCEEDED_THROTTLE | User has exceeded throttle threshold |
| USER_NOT_SUBSCRIBED_TO_PRODUCT | User not subscribed to specified product |
| SUBSCRIPTION_INACTIVE_OR_SUSPENDED | User subscription to product is Inactive or Suspended |
| VENDOR_SOFTWARE_INACTIVE | Vendor Software is Inactive |
| VENDOR_SOFTWARE_INVALID | Specified VendorSoftwareId is invalid or does not exist |
| SERVICE_NOT_AVAILABLE_IN_PRODUCT | User attempting to access service which is not present in their subscription |
| NO_SESSION | Session Token does not exist or has expired |
| TOO_MANY_REQUESTS | For each session/login/unique sessionToken the user can only make one call at a time. If two requests are sent simultaneously, requests will receive this message until previous requests are processed. If you want to send multiple requests, you must call login and use a second sessionToken sequence. |
| PRODUCT_REQUIRES_FUNDED_ACCOUNT | You cannot use this product unless your Betfair account is active. An active account is one in which your Account Statement page on Betfair.com contains at least one transaction. Transactions expire after three months. Transactions include depositing funds and placing a bet that is settled. |
| SERVICE_NOT_AVAILABLE_FOR_LOGIN_STATUS | You have logged in, but access to services is restricted to retrieveLIMBMessage, submitLIMBMessage, and logout. This may be because there is a login message that requires attention. |

Login (global)

The API Login service enables customers to log in to the API service and initiates a secure session for the user. Users can have multiple sessions 'alive' at any point in time.

Service Details

SOAP Action: login

Input

The following table describes the parameters for calling Login.

Table 6: 1 instance of LoginReq

| Parameter | Mandatory? | Type | Description |
|------------------|------------|---------|--|
| username | Y | String | The username with which to login to the API for a new session. |
| password | Y | String | The password with which to login to the API for a new session. |
| productId | Y | Integer | The API product ID with which to login to the API for a new session. If you want to use the Free Access API, use 82. If you are a paying API subscriber, use the Id provided when you signed up. |
| vendorSoftwareId | Y | Integer | The vendor software ID with which to login to the API for a new session. This is only relevant for software vendors and is provided when software vendors sign up. |
| locationId | Y | Integer | The location ID with which to login for a new session |
| ipAddress | Y | String | For applications that proxy the user's connection, the IP address of the user's computer. Betfair may inform you in the future if you need to provide this field, otherwise set this field to 0. |

Output

The following table describes the returned parameters for Login.

Table 7: 1 instance of LoginResp

| Parameter | Type | Notes |
|----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the session token. A session token is only returned for a successful login. |
| currency | String | The currency used by the account |
| errorCode | LoginErrorEnum | login specific error code (see enum below) |
| minorErrorCode | String | This field contains <code>age.verification.incomplete</code> if the user has not completed the age verification process with Betfair. |
| validUntil | Date | Logins will succeed until this date, after which logins will be rejected unless help desk is contacted. |

Error Code

The following table describes the error codes returned by Login. You should also check the response `minorErrorCode` field for additional information.

Table 8: LoginErrorEnum

| Value | Condition |
|---|---|
| OK | |
| ACCOUNT_CLOSED | Not used |
| ACCOUNT_SUSPENDED | Account suspended - Check for any <code>minorErrorCode</code> and inform the user to contact the Betfair helpdesk if appropriate. |
| API_ERROR | General API error. Check the APIResponseHeader <code>errorCode</code> field for additional information |
| FAILED_MESSAGE | The user cannot login until they acknowledge a message from Betfair (see Retrieve LIMB Message (global)). |
| INVALID_LOCATION | Invalid <code>LocationID</code> |
| INVALID_PRODUCT | Invalid <code>productID</code> entered |
| INVALID_USERNAME_OR_PASSWORD | Incorrect username and/or password supplied, or the account is closed or locked. |
| INVALID_VENDOR_SOFTWARE_ID | Invalid <code>vendorSoftwareId</code> supplied |
| LOGIN_FAILED_ACCOUNT_LOCKED | Not used |
| LOGIN_REQUIRE_TERMS_AND_CONDITIONS_ACCEPTANCE | Not used |
| LOGIN_RESTRICTED_LOCATION | Login origin from a restricted country |

| Value | Condition |
|-----------------------------------|--|
| LOGIN_UNAUTHORIZED | User has not been permissioned to use API login. |
| OK_MESSAGES | There are additional messages on your account. Please log in to the web site to view them. Note: If you get this message you have successfully logged in. |
| POKER_T_AND_C_ACCEPTANCE_REQUIRED | Account locked, Please login to the Betfair Poker website and assent to the terms and conditions |
| T_AND_C_ACCEPTANCE_REQUIRED | There are new Terms and Conditions. You can either use the RetrieveLIMBMessage and SubmitLIMBMessage services or accept the new terms through the Betfair website. Continued usage of the Betfair API and/or website will be considered acceptance of the new terms. |
| USER_NOT_ACCOUNT_OWNER | The specified account is not a trading account and therefore cannot be used for API access. |

**Note:**

Login service now takes a header within the SOAP request that contains the login parameters. Previously the SOAP header contained this information. There is now no need to modify the SOAP header.

Logout (global)

The API Logout service allows you to explicitly end your session.

Service Details

SOAP Action: `logout`

Input

The following table describes the parameters used for calling the logout service.

Table 9: 1 instance of LogoutReq

| Parameter | Mandatory? | Type | Description |
|-----------|------------|-------------------------|--|
| header | Y | <i>APIRequestHeader</i> | The API Request Header that contains the user's session token (see <i>Betfair Simple Data Types</i>). |

Output

The following table describes the parameters returned from the logout service.

Table 10: 1 Instance of LogoutResp

| Parameter | Type | Description |
|-----------|--------------------------|-------------------------|
| header | <i>APIResponseHeader</i> | The API Response Header |

| Parameter | Type | Description |
|----------------|---------------------------------|--|
| minorErrorCode | String | Reserved for future use. Currently always nil. |
| errorCode | LogoutErrorEnum | Service specific error code (see below). |

Error Codes

The following table describes the error codes returned by Logout.

Table 11: LogoutErrorEnum

| Value | Condition |
|-----------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |

KeepAlive (global)

The keep alive service can be used to stop a session timing out. Every call to the Betfair API returns a token, in the `sessionToken` field, that identifies a login session. Every time your application calls the Betfair API and is returned a `sessionToken`, the session timeout is reset to approximately 20 minutes. After the timeout has passed, the session is expired and you need to login again.

If you want to keep your login session active, but your application has not made any Betfair API calls that would generate a new `sessionToken` and reset the session timeout, you can call `keepAlive` to obtain a new `sessionToken` and reset the session timeout.

Service Details

SOAP Action: `keepAlive`

Input

The following table describes the

Table 12: 1 instance of KeepAliveRequest

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|--|
| header | Y | APIRequestHeader | The API Response Header that contains the user's session token (see Betfair Simple Data Types). |

parameters used for calling the KeepAlive service.

Output

The following table describes the parameters returned from the KeepAlive service.

Table 13: 1 instance of KeepAliveResp

| Parameter | Type | Notes |
|----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header |
| apiVersion | String | Not Used |
| minorErrorCode | String | Reserved for future use - currently always null |

**Note:**

Use this service to heartbeat the connection to the API server and prevent your session from timing out.

Read-Only Betting API Services Reference

Convert Currency (global)

The API ConvertCurrency service allows you to convert a currency, based on the Betfair currency exchange rate.

Service Details

SOAP Action: `convertCurrency`

Input

The following table describes the parameters used for calling the ConvertCurrency service.

Table 14: 1 instance of `ConvertCurrencyReq`

| Parameter | Mandatory? | Type | Description |
|--------------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| amount | Y | Double | Currency amount |
| fromCurrency | Y | String | Currency to be converted |
| toCurrency | Y | String | Desired currency |

Output

The following table describes the parameters returned from the ConvertCurrency service.

Table 15: 1 Instance of `ConvertCurrencyResp`

| Parameter | Type | Description |
|-----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| convertedAmount | Double | Converted amount in desired currency |
| errorCode | | Specific error code (see below) |

Error Codes

The following table describes the error codes returned by ConvertCurrency.

Table 16: ConvertCurrencyErrorEnum

| Value | Condition |
|-----------------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| CANNOT_CONVERT | Cannot convert the currency |
| INVALID_AMOUNT | Invalid currency amount supplied |
| INVALID_FROM_CURRENCY | The currency specified does not exist |
| INVALID_TO_CURRENCY | The currency specified does not exist |

Get Active Event Types (global)

The API GetActiveEventTypes service allows the customer to retrieve lists of all categories of sporting events (Games, Event Types) that are available to bet on: in other words, all those that have at least one currently active or suspended market associated with them. This means, therefore, that the service would, for example, always return the event types Soccer and Horse Racing but would not return Olympics 2004 or EURO 2004 after those events had finished.



Note:

The GetActiveEventTypes service is a global service, and it returns information about the events available on both the UK and the Australian exchange servers.

Service Details

SOAP Action: `getActiveEventTypes`

Input

The following table describes the parameters used for calling the GetActiveEventTypes service.

Table 17: 1 instance of GetActiveEventTypesReq

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|--|
| header | Y | APIRequestHeader | The API Response Header that contains the user's session token (see Betfair Simple Data Types). |
| locale | N | String | Specify the language for the reply if you want a different language than the account default. |

Output

The following table describes the parameters returned from the GetActiveEventTypes service.

Table 18: 1 Instance of GetEventTypesResp

| Parameter | Type | Description |
|----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| eventTypeItems | ArrayOfEventType | An array containing the active event types |
| minorErrorCode | String | Reserved for future use - currently always null |
| errorCode | see ... | Service specific error code |

The GetActiveEventTypes service returns an array of eventTypeItems. The table below shows the fields in EventType

Table 19: ArrayOfEventType 0 or more instances of EventType

| Parameter | Type | Description |
|--------------|---------|--|
| id | Integer | The ID of Event |
| name | String | The name of the Event Type |
| exchangeId | Integer | The Id of the exchange server that is hosting the betting markets for the eventType. This is returned only for “Horse Racing -Today’s Card” and “Greyhound - Today’s Card” events. For all other events, the result is zero. |
| nextMarketId | Integer | The market Id of the next race to start. This is returned only for “Horse Racing -Today’s Card” and “Greyhound - Today’s Card” events. For all other events, the result is zero. |

Error Codes

The following table describes the error codes returned by GetActiveEventTypes.

Table 20: GetEventsErrorEnum

| Value | Condition |
|------------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_EVENT_ID | Not used |

| Value | Condition |
|--------------------------------------|--|
| INVALID_LOCALE_DEFAULTING_TO_ENGLISH | The locale string was not recognized. Returned results are in English. |
| NO_RESULTS | There are no events to return |

Get All Currencies (global)

The API GetAllCurrencies service allows you to retrieve all the currencies (and their exchange rates in relation to GBP) that you can select to have your account data expressed in.

Service Details

SOAP Action: `getAllCurrencies`

Input

The following table describes the parameters used for calling the GetAllCurrencies service.

Table 21: 1 instance of `GetCurrenciesReq`

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |

Output

The following table describes the parameters returned from the GetAllCurrencies service.

Table 22: 1 Instance of `GetCurrenciesResp`

| Parameter | Type | Description |
|---------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| currencyItems | ArrayOfCurrency | An array of type Currency containing the currency code and exchange rate against GBP. |

The GetAllCurrencies service returns `currencyItems` as an array of Currency. The table below shows the fields in Currency

Table 23: ArrayOfCurrency 0 or more instances of Currency

| Parameter | Type | Description |
|--------------|--------|----------------------------|
| currencyCode | String | Three letter ISO 4217 code |

| Parameter | Type | Description |
|-----------|--------|--|
| rateGBP | Double | Exchange rate in GBP for the currency specified in the request |

Get All Currencies V2 (global)

The API GetAllCurrenciesV2 service allows you to retrieve an array of currencies supported by the Betfair Exchange, their exchange rates in relation to GBP, and the minimum bet amount for each currency.

Service Details

SOAP Action: `getAllCurrenciesV2`

Input

The following table describes the parameters used for calling the GetAllCurrenciesV2 service.

Table 24: 1 instance of `GetCurrenciesV2Req`

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |

Output

The following table describes the parameters returned from the GetAllCurrenciesV2 service.

Table 25: 1 Instance of `GetCurrenciesV2Resp`

| Parameter | Type | Description |
|---------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| currencyItems | ArrayOfCurrencyV2 | An array of type CurrencyV2 containing the currency code, exchange rate against GBP, and minimum stake information. |

The GetAllCurrenciesV2 service returns `currencyItems` as an array of CurrencyV2. The table below shows the fields in CurrencyV2

Table 26: ArrayOfCurrencyV2 0 or more instances of CurrencyV2

| Parameter | Type | Description |
|--------------|--------|----------------------------|
| currencyCode | String | Three letter ISO 4217 code |

| Parameter | Type | Description |
|------------------------|--------|--|
| rateGBP | Double | Exchange rate in GBP for the currency specified in the request |
| minimumStake | Double | The minimum bet amount in the currency |
| minimumRangeStake | Double | The minimum bet amount for range bets |
| minimumBSPLayLiability | Double | The minimum BSP Lay Liability amount for the currency |

Get All Event Types (global)

The API GetAllEventTypes service allows the customer to retrieve lists of all categories of sports (Games, Event Types) that have at least one market associated with them, regardless of whether that market is now closed for betting. This means that, for example, the service would always return the event types Soccer and Horse Racing and would also return Olympics 2004 or EURO 2004 for a certain period after the markets for those events had closed; it would also return Olympics 2004 or EURO 2004 for a certain period before the markets for those events had opened. The service returns information on future events to allow API programmers to see the range of events that will be available to bet on in the near future.



Note:

The GetAllEventTypes service is a global service, and it returns information about the events available on both the UK and the Australian exchange servers.

Service Details

SOAP Action: `getAllEventTypes`

The following table describes the parameters used for calling the GetAllEventTypes service.

Table 27: 1 instance of `GetEventTypesReq`

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| locale | N | String | Specify the language for the reply if you want a different language than the account default. For more information, |

Output

The following table describes the parameters returned from the GetAllEventTypes service.

Table 28: 1 Instance of `GetEventTypesResp`

| Parameter | Type | Description |
|----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| eventTypeItems | ArrayOfEventType | Array of event types |
| minorErrorCode | String | Reserved for future use - currently always null |
| errorCode | see ... | Service specific error codes. See below |

The `GetAllEventTypes` service returns `eventTypeItems` as an array. The table below shows the fields in `EventType`

Table 29: `ArrayOfEventType` 0 or more instances of `EventType`

| Parameter | Type | Description |
|--------------|------------------|---|
| id | Integer | The ID of Event |
| name | String | The name of the Event Type |
| nextMarketId | Integer | The market Id of the next race to start. This is returned only for “Horse Racing -Today’s Card” and “Greyhound - Today’s Card” events. For all other events, the result is zero. |
| exchangeId | ArrayOfEventType | If there is a market identified in the <code>nextMarketId</code> parameter, then the <code>exchangeId</code> parameter indicates the exchange server that is hosting that market. The possible values for the <code>exchangeId</code> parameter are ‘1’ for the UK exchange server, and ‘2’ for the Australian exchange server. |

Error Code

The following table describes the error codes returned by `GetAllEventTypes`.

Table 30: `GetEventsErrorEnum`

| Value | Condition |
|------------------|--|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader <code>errorCode</code> field for additional information |
| INVALID_EVENT_ID | Not used |

| Value | Condition |
|-------------------------------------|--|
| INVALID_LOCAL_DEFAULTING_TO_ENGLISH | The locale string was not recognized. Returned results are in English. |
| NO_RESULTS | No data available to return |

Get All Markets (exchange)

The API GetAllMarkets service allows you to retrieve information about all of the markets that are currently active or suspended on the given exchange. You can use this service to quickly analyse the available markets on the exchange, or use the response to build a local copy of the Betfair.com navigation menu. You can limit the response to a particular time period, country where the event is taking place, and event type. Otherwise, the service returns all active and suspended markets.

The return format is similar to the GetMarketPricesCompressed service.



Note:

You should develop your application such that it can easily handle future changes to the marketData format string, such as appended fields, returned by the GetAllMarkets service.

Service Details

SOAP Action: `getAllMarkets`

Known Issues

All BF Tennis, Golf, Cricket and Rugby Union markets are showing a prefix of 'Group A/B/C/D' in the menuPath, while there is no such valid component.

The market's Parent Event name in the menu path returned in the response to getMarket and getAllMarkets is sometimes truncated.

Input

The following table describes the parameters used for calling the GetAllMarkets service.

Table 31: 1 instance of GetAllMarketsReq

| Parameter | Mandatory? | Type | Description |
|--------------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token. <i>See “API Request/Response Header” on page 21.</i> |
| locale | N | String | The locale to use when returning results. If not specified, the default locale for the user's account is used. |
| eventTypeIds | N | ArrayOfInt | The events types to return. If not specified, markets |

| Parameter | Mandatory? | Type | Description |
|-----------|------------|--------------------|---|
| countries | N | ArrayOfCountryCode | from all event types are returned. The countries where the event is taking place as an array of ISO3 country codes. If not specified, markets from all countries (or international markets) for the specified exchange are returned. |
| fromDate | N | DateTime | If this is set, the response contains only markets where the market time is not before the specified date. A null value indicated no limit. |
| toDate | N | DateTime | If this is set, the response contains only markets where the market time is not after the specified date. A null value indicated no limit. |

Output

The following table describes the parameters returned from the GetAllMarkets service.

Table 32: 1 Instance of GetAllMarketsResp

| Parameter | Type | Description |
|----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token. <i>See “API Request/Response Header” on page 21.</i> |
| errorCode | see ... | Service specific error code |
| marketData | String | Delimited field containing market data (see below) |
| minorErrorCode | String | Reserved for future use - currently always null |

The GetAllMarkets service returns a string representing the market data. The table below shows the fields in marketData. Each market is delimited by a : (colon). Fields within a market are delimited by a ~ (tilde).

Table 33: marketData

| Field No. | Type | Name | Notes |
|-----------|----------|-----------------|---|
| 1 | Integer | Market ID | The Market Id of the market. Note: This is unique to an exchange, but is not guaranteed to be unique between exchanges. Always check the ExchangeID of the market. |
| 2 | String | Market name | The displayed name of the market. The name is returned in the account's default locale or in the locale specified in the request. |
| 3 | String | Market Type | The type of the market. |
| 4 | String | Market Status | The current status of the market (i.e., active or suspended). |
| 5 | DateTime | Event Date | The date and time the event starts (in milliseconds since January 1 1970 00:00:00 GMT) |
| 6 | String | Menu Path | The textual menu path for the market, not including the Market name at the end. The path is returned in the account's default locale or in the locale specified in the request. |
| 7 | String | Event Hierarchy | The full hierarchy of Event IDs leading to the specified market, including the final Market Id. |
| 8 | String | Bet Delay | The current bet delay. This will be non-zero when the market is in-play. Note: If you want to determine the moment a |

| Field No. | Type | Name | Notes |
|-----------|----------|----------------------|---|
| 9 | Integer | Exchange Id | market turns in-play, you should use <code>getMarketPricesCompressed</code> instead. The exchange Id for the market. 1 for the UK/Worldwide exchange, 2 for the Australian Exchange. |
| 10 | String | ISO3 Country Code | The Country Code for the event. An empty field indicates an international market that takes place in multiple countries. |
| 11 | DateTime | Last Refresh | The time (in milliseconds since January 1 1970 00:00:00 GMT) since the cached market data was last refreshed from the exchange database. The API caches market information for 5 minutes. |
| 12 | Integer | Number of Runners | The number of runners in the market |
| 13 | Integer | Number of Winners | The number of possible winners in the market |
| 14 | Double | Total Amount Matched | The total amount of money in GBP that have been matched on the specified market. |
| 15 | Boolean | BSP Market | If Y, indicates that the market supports Betfair Starting Price bets. |
| 16 | Boolean | Turning In Play | If Y, indicates that the market is scheduled to be turned in-play |

Example marketData output

```
:20158165~Match Odds~O~ACTIVE~1164223800000~\Soccer
\Scoottish Soccer\Bells League Div 1\Fixtures 22 November \Partick v
Clyde~/1/2695886/610072/10551708/10551709/20158165~0~1~GBR~1164192924479~3~1~8737.44~N~N:
```

**Note:**

The response may contain an ‘escape’ character, a backslash (\), to indicate that the subsequent character should be interpreted literally and not as a delimiter. For example, for markets with event times in the name, the response would look like: ...15\;35...

**Note:**

To recreate a textual menu that matches the numeric hierarchy, you must prepend the name for the event type (i.e., Soccer) and append the Market name returned (i.e., Partick v Clyde).

Error Code

The following table describes the error codes returned by GetAllMarkets.

Table 34: GetAllMarketsErrorEnum

| Value | Condition |
|-----------------------|--|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_COUNTRY_CODE | The country code submitted does not match a known country code. |
| INVALID_EVENT_TYPE_ID | The eventTypeid does not match any known eventTypes. See “Get All Event Types (global)” on page 36 |
| INVALID_LOCALE | The locale specified does not exist. See “Locale Specification” on page 211 |

Get Bet (exchange)

The API GetBet service allows you to retrieve information about a particular bet. Each request will retrieve all components of the desired bet.

**Note:**

You can retrieve Cancelled, Lapsed, and Voided bets from only settled markets and these bets are available for a maximum of 10 days from the date the market was settled.

Service Details

SOAP Action: getBet

Known Issues

The requestedSize field is always zero for voided or lapsed bets.

Bet prices of 1.01 or 1000 are returned for BSP Market on Close bets instead of 0.0. Ignore the contents of the price field when betCategoryType is M and the bet status is U.

The avgPrice is not returned for a matched BSP Market on Close bet when the market is suspended. Ignore the contents of s avgPrice when market status is 'suspended' and bet category type is 'market on close'. The avgPrice field should have the same value as the priceMatched field.

Incorrectly returns matches for unmatched and lapsed unmatched BSP bets. Ignore the matches section in the response when bet status is 'unmatched' and matched size is 0, or bet status is 'lapsed'.

Incorrectly returns an average price for lapsed bets. Ignore the contents of the avgPrice field, when bet status is 'lapsed'.

Input

The following table describes the parameters used for calling the GetBet service.

Table 35: 1 instance of GetBetReq

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| betId | Y | Long | The unique bet identifier |
| locale | N | String | Specify the language for the reply if you want a different language than the account default. For more information, |

Output

The following table describes the parameters returned from the GetBet service.

Table 36: 1 Instance of GetBetResp

| Parameter | Type | Description |
|-----------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| bet | Bet | The bet details |
| errorCode | see ... | Service specific API error. See below. |

The GetBet service returns bet as type Bet. The table below shows the fields in Bet

Table 37: 1 instance of Bet

| Parameter | Type | Description |
|-------------|---------|--|
| asianLineId | Integer | Id of the specific Asian handicap line |

| Parameter | Type | Description |
|--------------------|--|---|
| avgPrice | Double | The average matched price of the bet (null if no part has been matched) |
| betCategoryType | BetCategoryTypeEnum | Exchange, Market on Close SP bet, or Limit on Close SP bet. |
| betId | Long | Unique identifier generated for every bet placement |
| betPersistenceType | BetPersistenceTypeEnum | What happens to the bet when the market turns in-play. |
| betStatus | BetStatusEnum | Status of the bet. |
| betType | BetTypeEnum | Back or lay. |
| bspLiability | Double | The maximum liability for the BSP bet. |
| cancelledDate | dateTime | Date and time that the bet was cancelled (null if not applicable) |
| executedBy | String | This is used internally and always returns UNKNOWN |
| fullMarketName | String | The full name of the market. This string is localised to the language set in the language parameter, if set, otherwise to the user's language in their profile. |
| handicap | Double | The handicap selection when the asianLineId is > 0. You can use this to construct a unique string for the selection. I.e., "Chelsea + 3" |
| lapsedDate | dateTime | Date and time that the bet was lapsed (null if not applicable) |
| marketId | Integer | Id of the market |
| marketName | String | Name of the market |
| marketType | MarketTypeEnum | Market type data - please see Chapter 61 . |
| marketTypeVariant | MarketTypeVariantEnum | The Market variant for Asian Handicap markets |
| matchedDate | dateTime | Date and time that the bet was matched (null if not applicable) |
| matchedSize | Double | For an exchange bet, the amount matched |
| matches | ArrayOfMatch | If detailed is true, this will be the details of the matched portions of the bet |

| Parameter | Type | Description |
|---------------|----------|---|
| placedDate | dateTime | Date and time of bet placement |
| price | Double | Price of the remaining exchange bet |
| profitAndLoss | Double | Net result of bet |
| remainingSize | Double | Remaining unmatched, lapsed or cancelled amount of the bet |
| requestedSize | Double | Original stake amount of the exchange bet |
| selectionId | Integer | Id of the selection (this will be the same for the same selection across markets) |
| selectionName | String | Name of the selection |
| settledDate | dateTime | Date and time of bet settlement |
| voidedDate | dateTime | Date and time that the bet was voided (null if not applicable) |

Bet contains an array of `Match`. The table below shows the fields in `Match`

Table 38: ArrayOfMatch 0 or more instances of Match

| Parameter | Type | Description |
|---------------|----------------------|---|
| betStatus | <i>BetStatusEnum</i> | Status of the bet. For more information, see Chapter 61 . |
| matchedDate | dateTime | Date and time at the bet portion was matched |
| priceMatched | Double | Price at which this portion was matched |
| profitLoss | Double | Profit/loss on this bet portion (null for unsettled bets) |
| settledDate | dateTime | Date and time at the bet portion was settled (null for unsettled bets) |
| sizeMatched | Double | Size matched in this portion |
| transactionId | Integer | Unique identifier for the individual transaction |
| voidedDate | dateTime | Date and time that the bet was voided (null if not applicable) |

Error Codes

The following table describes the error codes returned by GetBet.

Table 39: GetBetErrorEnum

| Value | Condition |
|--------------------------------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| BET_ID_INVALID | Bet Id is invalid or does not exist. Make sure you have sent your request to the correct exchange server. The getBet service only searches on the exchange server it is sent to for the bet that you specify. |
| INVALID_LOCALE_DEFAULTING_TO_ENGLISH | The locale string was not recognized. Returned results are in English |
| MARKET_TYPE_NOT_SUPPORTED | Market Type is invalid or does not exist |
| NO_RESULTS | No results. |

Get Bet History (exchange)

The API GetBetHistory service allows you to retrieve information about the bets you have placed on a particular exchange. Each request can only retrieve bets of the same status (MATCHED/UNMATCHED etc.) Pagination through the result set is supported by means of the startRecord and recordCount parameters.



Note:

You can retrieve Cancelled, Lapsed, and Voided bets from only settled markets and these bets are available for a maximum of 10 days from the date the market was settled.

Service Details

SOAP Action: getBetHistory

Known Issues

Bet prices of 1.01 or 1000 are returned for BSP Market on Close bets instead of 0.0. Ignore the contents of the price field when betCategoryType is M and the bet status is U.

Incorrectly returns matches for unmatched and lapsed unmatched BSP bets. Ignore the matches section in the response when bet status is 'unmatched' and matched size is 0, or bet status is 'lapsed'.

Incorrectly returns an average price for lapsed bets. Ignore the contents of the avgPrice field, when bet status is 'lapsed'.

Input

The following table describes the parameters used for calling the GetBetHistory service.

Table 40: 1 instance of GetBetHistoryReq

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |

| Parameter | Mandatory? | Type | Description |
|------------------|------------|----------------------|--|
| betTypesIncluded | Y | <i>BetStatusEnum</i> | Indicates the status of the bets to include in the response. If your betHistory request is for a specific market (in other words, if you have specified a marketId in your request), then you must specify either M or U as the value for betTypesIncluded. Otherwise you will receive an INVALID_BET_STATUS error. Only settled markets return cancelled, void, or lapsed bets. |
| detailed | Y | Boolean | Whether to show details of all the matches on a single bet |
| eventTypeIds | Y | ArrayOfInt | Event types to return. For matched and unmatched bets only, you can submit an empty array, “<eventTypeIds></eventTypeIds>”, and specify zero as the marketId to receive records of all your bets on the exchange. . |
| marketId | N | Integer | Returns the records of your matched or unmatched bets for the specified market. If you use this parameter you must submit an empty eventTypeId array “<eventIds></eventIds>” Note that, if you specify a marketId, you must also specify either M or U as the value for the betTypesIncluded |

| Parameter | Mandatory? | Type | Description |
|---------------------|------------|---|---|
| locale | N | String | parameter (see above in this table).. Specify the locale for the reply if you want a different language than the account default. For more information, see “Locale Specification” on page 211 |
| timezone | N | String | Specify an alternative time-zone from the user account default |
| marketTypesIncluded | Y | ArrayOf MarketTypeEnum | Indicates the types of market that you want your betting history returned for. For the valid values, see Chapter 61 . |
| placedDateFrom | Y | Date | Return records on or after this date. (The date data type is pre-defined in XML. For its format, check: http://www.w3.org/TR/xmlschema-2/#date .) |
| placedDateTo | Y | Date | Return records on or before this date. (The date data type is pre-defined in XML. For its format, check: http://www.w3.org/TR/xmlschema-2/#date .) |
| recordCount | Y | Integer | The maximum number of records to return. This number must be between 1 and 100, inclusive. |
| sortBetsBy | Y | BetsOrderByEnum | How the bets are ordered. For a list of the valid values, see Chapter 61 . |
| startRecord | Y | Integer | The first record number to return (supports paging). Record numbering |

| Parameter | Mandatory? | Type | Description |
|-----------|------------|------|--|
| | | | starts from 0. For example, to retrieve the third record and higher, set startRecord to 2. |

The following table shows valid eventTypeId, MarketId and BetStatus combinations.

Table 41: Valid eventTypeIds, MarketId, and betTypesIncluded combinations

| betTypesIncluded | eventTypeIds Specified | marketId Specified | Both Specified |
|-----------------------------|------------------------|--------------------|----------------|
| Matched or Unmatched | Optional | Optional | Not allowed |
| Settled, Cancelled, or Void | Required | Not allowed | Not allowed |

Output

The following table describes the parameters returned from the GetBetHistory service.

Table 42: 1 Instance of GetBetHistoryResp

| Parameter | Type | Description |
|------------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| betHistoryItems | ArrayOfBet | An array of Bet containing details of each bet placed. |
| errorCode | see ... | Specific error code (see enum below) |
| minorErrorCode | String | Reserved for future use - currently always null |
| totalRecordCount | Integer | Total number of records in the request (less than or equal to maxRecords) |

The GetBetHistory service returns betHistoryItems as an array of Bet. The table below shows the fields in Bet

Table 43: 1 instance of Bet

| Parameter | Type | Description |
|-------------|---------|---|
| asianLineId | Integer | Id of the specific Asian handicap line |
| avgPrice | Double | The average matched price of the bet (null if no part has been matched) |

| Parameter | Type | Description |
|--------------------|--|---|
| betCategoryType | BetCategoryTypeEnum | Exchange, Market on Close SP bet, or Limit on Close SP bet. |
| betId | Integer | Unique identifier generated for every bet placement |
| betPersistenceType | BetPersistenceTypeEnum | What happens to the bet when the market turns in-play. |
| betStatus | BetStatusEnum | Status of the bet. |
| betType | BetTypeEnum | Back or lay. |
| bspLiability | Double | The maximum liability for the BSP bet. |
| cancelledDate | dateTime | Date and time that the bet was cancelled (null if not applicable) |
| executedBy | String | This is used internally and always returns UNKNOWN |
| fullMarketName | String | The full name of the market. This string is localised to the language set in the language parameter, if set, otherwise to the user's language in their profile. |
| handicap | Double | The handicap selection when the asianLineId is > 0. You can use this to construct a unique string for the selection. I.e., "Chelsea + 3" |
| lapsedDate | dateTime | Date and time that the bet was lapsed (null if not applicable) |
| marketId | Integer | Id of the market |
| marketName | String | Name of the market |
| marketType | MarketTypeEnum | Market type data - please see Chapter 61 . |
| marketTypeVariant | MarketTypeVariantEnum | The Market variant for Asian Handicap markets |
| matchedDate | dateTime | Date and time that the bet was matched (null if not applicable) |
| matchedSize | Double | For an exchange bet, the amount matched |
| matches | ArrayOfMatch | If detailed is true, this will be the details of the matched portions of the bet |
| placedDate | dateTime | Date and time of bet placement |
| price | Double | Price of the remaining exchange bet |

| Parameter | Type | Description |
|---------------|----------|---|
| profitAndLoss | Double | Net result of bet |
| remainingSize | Double | Remaining unmatched, lapsed or cancelled amount of the bet |
| requestedSize | Double | Original stake amount of the exchange bet |
| selectionId | Integer | Id of the selection (this will be the same for the same selection across markets) |
| selectionName | String | Name of the selection |
| settledDate | dateTime | Date and time of bet settlement |
| voidedDate | dateTime | Date and time that the bet was voided (null if not applicable) |

Bet contains an array of `Match`. The table below shows the fields in `Match`

Table 44: ArrayOfMatch 0 or more instances of Match

| Parameter | Type | Description |
|---------------|----------------------|---|
| betStatus | <i>BetStatusEnum</i> | Status of the bet. For more information, see Chapter 61 . |
| matchedDate | dateTime | Date and time that the bet portion was matched |
| priceMatched | Double | Price at which this portion was matched |
| profitLoss | Double | Profit/loss on this bet portion (null for unsettled bets) |
| settledDate | dateTime | Date and time that the bet portion was settled (null for unsettled bets) |
| sizeMatched | Double | Size matched in this portion |
| transactionId | Integer | Unique identifier for the individual transaction |
| voidedDate | dateTime | Date and time that the bet was voided (null if not applicable) |

Error Code

The following table describes the error codes returned by `GetBetHistory`.

Table 45: GetBetHistoryErrorEnum

| Value | Condition |
|-----------|---|
| OK | |
| API_ERROR | General API error. Check the <i>APIResponseHeader</i> <code>errorCode</code> field for additional information |

| Value | Condition |
|-------------------------------------|---|
| INVALID_BET_STATUS | Either you have not supplied a betStatus or you have supplied an invalid betStatus. Note that, if your betHistory request is for a specific market (in other words, if you have specified a marketId in your request), then you must specify either M or U as the value for betTypesIncluded. Otherwise you will receive an INVALID_BET_STATUS error. |
| INVALID_EVENT_TYPE_ID | Event Types not supplied |
| INVALID_LOCAL_DEFAULTING_TO_ENGLISH | The language string was not recognised |
| INVALID_MARKET_TYPE | Market Types is not supplied |
| INVALID_ORDER_BY | Order is not supplied |
| INVALID_RECORD_COUNT | Max Records < 0 or > 100 |
| INVALID_START_RECORD | Start record is not supplied or is invalid |
| NO_RESULTS | No bets meet the specified criteria. It is possible that you have sent your request to the wrong exchange server. For example, if you have never placed a bet on the Australian exchange server, and you then send it a getBetHistory request to it, you will receive this error message. |

Get Bet Lite (exchange)

The API GetBetLite service allows you to retrieve information about a bet. This is the lite version of the GetBet service.

Service Details

SOAP Action: getBetLite

Input

The following table describes the parameters used for calling the GetBetLit service.

Table 46: 1 instance of GetBetLiteReq

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see API Request/Response Header). |
| betId | Y | Long | The unique bet identifier |

Output

The following table describes the parameters returned from the GetBetLite service.

Table 47: 1 Instance of `GetBetLiteResp`

| Parameter | Type | Description |
|-----------|--|--|
| header | <i>APIResponseHeader</i> | The API Response Header that contains the session token (see <i>API Request/Response Header</i>). |
| betLite | <i>see ...</i> | The bet details |
| errorCode | <i>see ...</i> | Service specific API error. See below. |

The `GetBetLite` service returns `betLite` as type `BetLite`. The table below shows the fields in `BetLite`

Table 48: `BetLite`

| Parameter | Type | Description |
|--------------------|---|---|
| betCategoryType | <i>BetCategoryTypeEnum</i> | Exchange, Market on Close SP bet, or Limit on Close SP bet. |
| betId | Long | Unique identifier generated for every bet placement |
| betPersistenceType | <i>BetPersistenceTypeEnum</i> | What happens to the bet when the market turns in-play. |
| betStatus | <i>BetStatusEnum</i> | Status of the bet. |
| bspLiability | Double | The maximum liability for the BSP bet. |
| marketId | Integer | Id of the market |
| matchedSize | Double | The amount matched |
| remainingSize | Double | Remaining unmatched, lapsed or cancelled amount of the bet |

Error Codes

The following table describes the error codes returned by `GetBetLite`.

Table 49: `GetBetErrorEnum`

| Value | Condition |
|----------------|--|
| OK | |
| API_ERROR | General API error. Check the <i>APIResponseHeader</i> <code>errorCode</code> field for additional information |
| BET_ID_INVALID | Bet Id is invalid or does not exist. Make sure you have sent your request to the correct exchange server. The <code>getBetLite</code> service only searches on the exchange server it is sent to for the bet that you specify. |

| Value | Condition |
|--------------------------------------|---|
| INVALID_LOCALE_DEFAULTING_TO_ENGLISH | The locale string was not recognized. Returned results are in English |
| MARKET_TYPE_NOT_SUPPORTED | Market Type is invalid or does not exist |
| NO_RESULTS | No results. |

Get Bet Matches Lite (exchange)

The API GetBetMatchesLite service allows you to retrieve information about a the matched portion of a bet. This is a lite version of the GetBet service that returns information on matched bets.

Service Details

SOAP Action: `getBetMatchesLite`

Input

The following table describes the parameters used for calling the GetBetMatchesLite service.

Table 50: 1 instance of `GetBetMatchesLiteReq`

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| betId | Y | Long | The unique bet identifier |

Output

The following table describes the parameters returned from the GetBetMatchesLite service.

Table 51: 1 Instance of `GetBetMatchesLiteResp`

| Parameter | Type | Description |
|------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| matchLites | ArrayOfMatchLite | The bet details. |
| errorCode | see ... | Service specific API error. See below. |

The GetBetMatchesLite service returns an array of matchLite as type MatchLite. The table below shows the fields in MatchLite.

Table 52: ArrayOfMatchLite 0 or more instances of MatchLite

| Parameter | Type | Description |
|---------------|-------------------------------|--|
| betStatus | BetStatusEnum | Status of the bet. For more information, see Betfair Simple Data Types . |
| matchedDate | dateTime | Date and time at the bet portion was matched |
| priceMatched | Double | Price at which this portion was matched |
| sizeMatched | Double | Size matched in this portion |
| transactionId | Long | Unique identifier for the individual transaction |

Error Codes

The following table describes the error codes returned by GetBetMatchesLite.

Table 53: GetBetErrorEnum

| Value | Condition |
|---------------------------|--|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| BET_ID_INVALID | Bet Id is invalid or does not exist. Make sure you have sent your request to the correct exchange server. The getBetMatchesLite service only searches on the exchange server it is sent to for the bet that you specify. |
| MARKET_TYPE_NOT_SUPPORTED | Market Type is invalid or does not exist |
| NO_RESULTS | No results. |

Get Complete Market Prices Compressed (exchange)

The API GetCompleteMarketPricesCompressed service allows you to retrieve all back and lay stakes for each price on the exchange for a given Market ID in a compressed format. The information returned is similar to the GetDetailAvailableMarketDepth, except it returns the data for an entire market, rather than just one selection.

Service Details

SOAP Action: `getCompleteMarketPricesCompressed`

Known Issues

If the optional currencyCode is not specified, getCompleteMarketPricesCompressed and getMarketTradeVolumeCompressed occasionally return prices using an incorrect currency. The workaround is to always pass the desired currency in the currencyCode field.

Input

The following table describes the parameters used for calling the GetCompleteMarketPricesCompressed service.

Table 54: 1 instance of GetCompleteMarketPricesCompressedReq

| Parameter | Mandatory? | Type | Description |
|--------------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| currencyCode | N | String | The three letter ISO 4217 code. If not supplied, user's currency is used |
| marketId | Y | Integer | ID of the desired market |

Output

The following table describes the parameters returned from the GetCompleteMarketPricesCompressed service.

Table 55: 1 Instance of GetMarketPricesCompressedResp

| Parameter | Type | Description |
|----------------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header. |
| errorCode | see ... | Specific error code (see enum below) |
| completeMarketPrices | String | Delimited field containing market price records (see below) |
| currencyCode | String | The currency used in the response |
| minorErrorCode | String | Reserved for future use - currently always null |

The GetCompleteMarketPricesCompressed service returns a string representing the compressed market prices. The table below shows the fields in

Table 56: marketPrices

| Field No. | Type | Name | Notes |
|----------------------------|---------|---------------|--|
| 1 | Integer | Market ID | The market Id of the market |
| 2 | Integer | In Play Delay | The number of seconds delay between submission and a bet actually getting placed. This is greater than 0 if and only if the market is in-play. |
| Removed runner information | | | |

| Field No. | Type | Name | Notes |
|--|---------|----------------------|--|
| fields delimited by , (commas) then delimited by a ; (semicolon) between each runner, 0 or more iterations | | | |
| 1 | String | Selection Name | The name of the removed runner |
| 2 | Time | Removed Time | The time the runner was removed in the format "H.mm" where H is the hour of the day (0 - 23) and mm is the minute of the hour (00 - 59). |
| 3 | String | Reduction Factor | The reduction factor applied to the remaining runners |
| Runner information fields then delimited by a : (colon), 0 or more iterations | | | |
| 1 | Integer | Selection ID | Id of the selection (this will be the same for the same selection across markets) |
| 2 | Integer | Order Index | The order in which the items are displayed on Betfair |
| 3 | Double | Total Amount Matched | The total amount matched on this selection (regardless of price) |
| 4 | Double | Last Price Matched | The last price at which a selection was matched |
| 5 | Double | Handicap | Handicap of the market (applicable to Asian handicap markets) |

| Field No. | Type | Name | Notes |
|--|---------|-----------------------|---|
| 6 | Double | Reduction Factor | Reduction in the odds that applies in case this runner does not participate |
| 7 | Boolean | Vacant | Used to indicate a Vacant Trap for withdrawn runners in Greyhound markets |
| 8 | Integer | Asian Line Id | The Asian Line Id for Asian Handicap markets |
| 9 | Double | Far SP Price | A prediction of the eventual starting price. 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 exchange at SP. |
| 10 | Double | Near SP Price | A prediction of the eventual starting price. 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. |
| 11 | Double | Actual SP Price | The actual Betfair Starting Price. Before the starting price is determined, this field contains zero. |
| Runner Prices then delimited by a (pipe), 0 or more iterations (1 iteration per price) | | | |
| 1 | Double | Price | Odds |
| 2 | Double | Back Amount Available | The total available to back at the odds specified |

| Field No. | Type | Name | Notes |
|-----------|--------|---------------------------------|---|
| 3 | Double | Lay Amount Available | The total available to lay at the odds specified |
| 4 | Double | Total BSP Back available amount | The BSP available to back amount. 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. Post-reconciliation, this field is zero for all prices |
| 5 | Double | Total BSP Lay available amount | The BSP available to lay amount. 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. Post-reconciliation, this field is zero for all prices. |

Example completeMarketPrices output

```
20771785~0~:58805~3~11510.72~4.1~~~false~~~~~|
1.01~673.36~0.0~0.0~0.0~1.02~6.75~0.0~0.0~0.0~1.05~2.25~0.0~0.0~0.0~1.1~1.5~0.0~0.0~
1.01~679.2~0.0~0.0~0.0~1.02~3.0~0.0~0.0~0.0~1.05~2.25~0.0~0.0~0.0~1.1~1.5~0.0~0.0~
1.01~676.1~0.0~0.0~0.0~1.02~3.0~0.0~0.0~0.0~1.05~2.25~0.0~0.0~0.0~1.1~1.5~0.0~0.0~
```

**Note:**

The response may contain an ‘escape’ character, a backslash (\), to indicate that the subsequent character should be interpreted literally and not as a delimiter. For example, in the Market Info field, you may see the following: NR\:(AEST).

Type Double may contain numbers, INF, -INF, and NaN.

Error Code

The following table describes the error codes returned by GetCompleteMarketPricesCompressed.

Table 57: GetCompleteMarketPricesErrorEnum

| Value | Condition |
|-----------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |

| Value | Condition |
|---------------------------|---|
| INVALID_CURRENCY | Currency code not a valid 3 letter ISO 4217 currency abbreviation |
| INVALID_MARKET | Market ID is not a valid market id. Check that you have sent your service request to the correct exchange server (the Australian exchange server cannot see markets on the UK exchange server, and vice versa). |
| MARKET_TYPE_NOT_SUPPORTED | The MarketID supplied refers to a market that is not supported by the API. |
| EVENT_CLOSED | The market has closed. |
| EVENT_SUSPENDED | The market is suspended |
| EVENT_INACTIVE | The market is inactive |

Get Current Bets (exchange)

The API GetCurrentBets service allows you to retrieve information about your current bets on a particular exchange server. Information can be retrieved for either a single betting market or for all markets (on the exchange server) that you have current bets placed in.

This requests supports paging through the result set by means of the `startRecord` and `recordCount` parameters. You should use the paging feature of this call only for bets where the result set does not change over time, such as settled bets.

Where the status of the bet may change between two calls (such as unmatched bets), the `getCurrentBets` service does not guarantee to page properly through the results. To page through matched and unmatched bets where the results are subject to change over time, use the `getMUBets` service.



Note:

Betfair recommends that you use the newer `GetMUBets` service instead of `GetCurrentBets`.

Service Details

SOAP Action: `getCurrentBets`

Known Issues

If a `marketId` is not supplied, `orderBy` is ignored except when `MATCHED_DATE` is specified. Without a `marketId`, all other `orderBy` types are ignored.

Bet prices of 1.01 or 1000 are returned for BSP Market on Close bets instead of 0.0. Ignore the contents of the price field when `betCategoryType` is M and the bet status is U.

The `avgPrice` is not returned for a matched BSP Market on Close bet when the market is suspended. Ignore the contents of `s avgPrice` when market status is 'suspended' and bet category type is 'market on close'. The `avgPrice` field should have the same value as the `priceMatched` field.

Input

The following table describes the parameters used for calling the `GetCurrentBets` service.

Table 58: 1 instance of `GetCurrentBetsReq`

| Parameter | Mandatory? | Type | Description |
|-------------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| betStatus | Y | BetStatusEnum | If MarketID is supplied, bet status must be U or M- see Betfair Simple Data Types for details. |
| detailed | Y | Boolean | If true, returns an array containing details of the matched portions of bets. |
| locale | N | String | Specify the language for the reply if you want a different language than the account default. For more information, see Locale Specification . |
| timezone | N | String | If a timezone is specified and the fullMarketName field contains an event time, the event time is adjusted to the specified timezone. |
| marketId | N | Integer | Specify the market for which current bets are required. Alternatively, you can leave the market Id blank. If you do not supply it, the API returns all your current bets on the Exchange. |
| orderBy | Y | BetsOrderByEnum | If Ordering is not NONE - please see Betfair Simple Data Types . |
| recordCount | Y | Integer | Maximum number of records to return |

| Parameter | Mandatory? | Type | Description |
|--------------------|------------|---------|--|
| startRecord | Y | Integer | The first record number to return (supports paging). Record numbering starts from 0. For example, to retrieve the third record and higher, set startRecord to 2. |
| noTotalRecordcount | Y | Boolean | Specify true to prevent the API from counting the total number of records (bets) that it returns. If you specify true, the totalRecordCount output parameter will contain the value 0. If the API does not perform a record count, it will return a reply to your GetCurrentBets request more quickly. |

The following table shows the valid ordering and bet status.

Table 59: Valid Ordering, Bet status combinations

| | Unmatch | Matched | Settled | Cancelled | Lapsed | Voided |
|----------------|---------|---------|---------|-----------|--------|--------|
| NONE | Y | Y | | Y | Y | Y |
| BET_ID | | | | | | |
| MARKET_NAME | | | | | | |
| PLACED_DATE | | Y | | | | |
| MATCHED_DATE | | Y | | | | |
| CANCELLED_DATE | | | | Y | | |

Output

The following table describes the parameters returned from the GetCurrentBets service.

Table 60: 1 Instance of GetCurrentBetsResp

| Parameter | Type | Description |
|-----------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| bets | ArrayOfBet | The Bet information |

| Parameter | Type | Description |
|------------------|-------------------------|--|
| errorCode | see ... | Specific error code (see enum below) |
| minorErrorCode | String | Reserved for future use - currently always null |
| totalRecordCount | Integer | Total number of records available. (If you specified true for the noTotalRecordCount input parameter in your GetCurrentBets request, the value of this parameter will be 0.) |

The GetCurrentBets service returns `bets` as an array of `Bet`. The table below shows the fields in `Bet`

Table 61: ArrayOfBet 0 or more instances of Bet

| Parameter | Type | Description |
|--------------------|--|---|
| asianLineId | Integer | Id of the specific Asian handicap line |
| avgPrice | Double | The average matched price of the bet (null if no part has been matched) |
| betId | Long | Unique identifier generated for every bet placement |
| betStatus | BetStatusEnum | Status of the bet. |
| betType | BetTypeEnum | Back or lay. |
| betCategoryType | BetCategoryTypeEnum | Exchange, Market on Close SP bet, or Limit on Close SP bet. |
| betPersistenceType | BetPersistenceTypeEnum | What happens to the bet when the market turns in-play. |
| cancelledDate | dateTime | Date and time that the bet was cancelled (null if not applicable) |
| lapsedDate | dateTime | Date and time that the bet was lapsed (null if not applicable) |
| marketId | Integer | Id of the market |
| marketName | String | Name of the market |
| fullMarketName | String | The full name of the market. This string is localised to the language set in the language parameter, if set, otherwise to the user's language in their profile. |
| marketType | MarketTypeEnum | Market type data - please see Chapter 61 . |

| Parameter | Type | Description |
|-------------------|------------------------------|--|
| matchedDate | dateTime | Date and time that the bet was matched (null if not applicable) |
| matchedSize | Double | For an exchange bet, the amount matched. Only valid when betStatus is M. If betStatus is U, this field is always 0.0. |
| matches | ArrayOfMatch | If detailed is true, this will be the details of the matched portions of the bet |
| placedDate | dateTime | Date and time of bet placement |
| bspLiability | Double | The maximum liability for the BSP bet. |
| price | Double | Price of the remaining exchange bet |
| profitAndLoss | Double | Net result of bet |
| selectionId | Integer | Id of the selection (this will be the same for the same selection across markets) |
| selectionName | String | Name of the selection |
| settledDate | dateTime | Date and time of bet settlement |
| remainingSize | Double | Remaining unmatched, lapsed or cancelled amount of the bet. Only valid when betStatus is U. If betStatus is M, this field is always 0.0. |
| requestedSize | Double | Original stake amount of the exchange bet |
| voidedDate | dateTime | Date and time that the bet was voided (null if not applicable) |
| executedBy | String | This is used internally and always returns UNKNOWN |
| handicap | Double | The handicap selection when the asianLineId is > 0. You can use this to construct a unique string for the selection. I.e., "Chelsea + 3" |
| marketTypeVariant | <i>MarketTypeVariantEnum</i> | The Market variant for Asian Handicap markets |

Bet contains an array of Match. The table below shows the fields in Match

Table 62: ArrayOfMatch 0 or more instances of Match

| Parameter | Type | Description |
|---------------|-------------------------------|---|
| betStatus | BetStatusEnum | Status of the bet. For more information, please see Betfair Simple Data Types . |
| matchedDate | dateTime | Date and time that the bet portion was matched |
| priceMatched | Double | Price at which this portion was matched |
| profitLoss | Double | Profit/loss on this bet portion (null for unsettled bets) |
| settledDate | dateTime | Date and time that the bet portion was settled (null for unsettled bets) |
| sizeMatched | Double | Size matched in this portion |
| transactionId | Integer | Unique identifier for the individual transaction |
| voidedDate | dateTime | Date and time that the bet was voided (null if not applicable) |

Error Codes

The following table describes the error codes returned by GetCurrentBets.

Table 63: GetCurrentBetsErrorEnum

| Value | Condition |
|-------------------------------|--|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_BET_STATUS | Market ID is not present and Bet Status is MU. |
| INVALID_BET_STATUS_FOR_MARKET | Market ID is present and Bet Status is VOIDED, LAPSED or CANCELLED |
| INVALID_MARKET_ID | Market ID is negative or does not exist. Check that you have sent your service request to the correct exchange server (the Australian exchange server cannot see markets on the UK exchange server, and vice versa). |
| INVALID_ORDER_BY_FOR_STATUS | Ordering is not NONE and: <ol style="list-style-type: none"> 1. Bet Status is MATCHED and Ordering is neither MATCHED_DATE or PLACED_DATE 2. Bet Status is UNMATCHED and ordering isn't PLACED_DATE 3. Bet Status is LAPSED or VOIDED and Ordering is not PLACED_DATE |

| Value | Condition |
|----------------------|--|
| | Bet Status is CANCELLED and Ordering is not CANCELLED_DATE |
| INVALID_RECORD_COUNT | Record Count is negative |
| INVALID_START_RECORD | Start record is not supplied or is invalid |
| NO_RESULT | |

Get Current Bets Lite (exchange)

The API GetCurrentBetsLite service allows you to retrieve information about your current bets on a particular exchange. Information can be retrieved for either a single betting market or for all markets (on the exchange) that you have current bets placed in.

This requests supports paging through the result set by means of the `startRecord` and `recordCount` parameters. You should use the paging feature of this call only for bets where the result set does not change over time, such as settled bets.

Where the status of the bet may change between two calls (such as unmatched bets), the `getCurrentBetsLite` service does not guarantee to page properly through the results. To page through matched and unmatched bets where the results are subject to change over time, use the `getMUBets` service.

Service Details

SOAP Action: `getCurrentBetsLite`

Input

The following table describes the parameters used for calling the `getCurrentBetsLite` service.

Table 64: 1 instance of `GetCurrentBetsLiteReq`

| Parameter | Mandatory? | Type | Description |
|------------------------|------------|----------------------------------|---|
| <code>header</code> | Y | APIRequestHeader | The API Request Header that contains the session token (see API Request/Response Header). |
| <code>betStatus</code> | Y | BetStatusEnum | If MarketID is supplied, bet status must be U or M - see Betfair Simple Data Types for details. |
| <code>marketId</code> | N | Integer | Specify the market for which current bets are required. Alternatively, you can leave the market Id blank. If you do not supply it, the API returns all your current bets on the particular exchange |

| Parameter | Mandatory? | Type | Description |
|--------------------|------------|------------------------|--|
| orderBy | Y | <i>BetsOrderByEnum</i> | server that you send the request to. If Ordering is not NONE - please see Chapter 61 . |
| recordCount | Y | Integer | Maximum number of records to return |
| startRecord | Y | Integer | The first record number to return (supports paging). Record numbering starts from 0. For example, to retrieve the third record and higher, set startRecord to 2. |
| noTotalRecordcount | Y | Boolean | Specify true to prevent the API from counting the total number of records (bets) that it returns. If you specify true, the totalRecordCount output parameter will contain the value 0. If the API does not perform a record count, it will return a reply to your GetCurrentBetsLite request more quickly. |

The following table shows the allowed ordering for each Bet status.

Table 65: Allowed Ordering, Bet status combinations

| | Unmatch | Matched | Settled | Cancelled | Lapsed | Voided |
|----------------|---------|---------|---------|-----------|--------|--------|
| NONE | Y | Y | | Y | Y | Y |
| BET_ID | | | | | | |
| MARKET_NAME | | | | | | |
| PLACED_DATE | | Y | | | | |
| MATCHED_DATE | | Y | | | | |
| CANCELLED_DATE | | | | Y | | |

Output

The following table describes the parameters returned from the GetCurrentBetsLite service.

Table 66: 1 Instance of `GetCurrentBetsLiteResp`

| Parameter | Type | Description |
|------------------|--|--|
| header | <i>APIResponseHeader</i> | The API Response Header that contains the updated session token (see <i>Betfair Simple Data Types</i>). |
| betLites | ArrayOfBetLite | The Bet information |
| errorCode | <i>see ...</i> | Specific error code (see enum below) |
| minorErrorCode | String | Reserved for future use - currently always null |
| totalRecordCount | Integer | Total number of records available. (If you specified true for the noTotalRecordCount input parameter in your GetCurrentBetsLite request, the value of this parameter will be 0.) |

The GetCurrentBetsLite service returns `betLites` as an array of `BetLite`. The table below shows the fields in `BetLite`

Table 67: ArrayOfBetLite 0 or more instances of `BetLite`

| Parameter | Type | Description |
|--------------------|---|---|
| betCategoryType | <i>BetCategoryTypeEnum</i> | Exchange, Market on Close SP bet, or Limit on Close SP bet. |
| betId | Long | Unique identifier generated for every bet placement |
| betPersistenceType | <i>BetPersistenceTypeEnum</i> | What happens to the bet when the market turns in-play. |
| betStatus | <i>BetStatusEnum</i> | Status of the bet. |
| bspLiability | Double | The maximum liability for the BSP bet. |
| marketId | Integer | Id of the market |
| matchedSize | Double | The amount matched |
| remainingSize | Double | Remaining unmatched, lapsed or cancelled amount of the bet |

Error Codes

The following table describes the error codes returned by GetCurrentBetsLite.

Table 68: `GetCurrentBetsErrorEnum`

| Value | Condition |
|-------|-----------|
| OK | |

| Value | Condition |
|-------------------------------|---|
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_BET_STATUS | Starting record is negative |
| INVALID_BET_STATUS_FOR_MARKET | Market ID is present and Bet Status is VOIDED, LAPSED or CANCELLED |
| INVALID_MARKET_ID | Market ID is negative or does not exist. Check that you have sent your service request to the correct exchange server (the Australian exchange server cannot see markets on the UK exchange server, and vice versa). |
| INVALID_ORDER_BY_FOR_STATUS | Ordering is not NONE and: <ol style="list-style-type: none"> 1. Bet Status is MATCHED and Ordering is neither MATCHED_DATE or PLACED_DATE 2. Bet Status is UNMATCHED and ordering isn't PLACED_DATE 3. Bet Status is LAPSED or VOIDED and Ordering is not PLACED_DATE Bet Status is CANCELLED and Ordering is not CANCELLED_DATE |
| INVALID_RECORD_COUNT | Record Count is negative |
| INVALID_START_RECORD | Start record is not supplied or is invalid |
| NO_RESULT | There were no bets that matched the criteria |

Get Detail Available Market Depth (exchange)

The API GetDetailAvailableMarketDepth service returns the current odds and available back/lay amounts on a runner in an event.



Note:

Betfair recommends that you use the new GetCompleteMarketPricesCompressed service instead. The GetCompleteMarketPricesCompressed services returns the market depth details for all runners in a single response.

Service Details

SOAP Action: `getDetailAvailableMktDepth`

Input

The following table describes the parameters used for calling the GetDetailAvailableMktDepth service.

Table 69: 1 instance of `GetDetailedAvailMktDepthReq`

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session |

| Parameter | Mandatory? | Type | Description |
|--------------|------------|---------|--|
| currencyCode | Y | String | token (see Betfair Simple Data Types). Three letter ISO 4217 code. If not supplied, users currency is used |
| asianLineId | N | Integer | Mandatory if the market specified by Market ID is an Asian Market, otherwise optional |
| locale | N | String | Specify the language for the reply if you want a different language than the account default |
| marketId | Y | Integer | The desired market |
| selectionId | Y | Integer | The desired runner id |

Output

The following table describes the parameters returned from the GetDetailAvailableMktDepth service.

Table 70: 1 Instance of GetDetailedAvailMktDepthResp

| Parameter | Type | Description |
|----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| errorCode | see ... | Specific error code (see enum below) |
| minorErrorCode | String | Reserved for future use - currently always null |
| priceItems | ArrayOfAvailabilityInfo | Array of odds and total amounts showing the depth in the market for the specified selection/runner. |

The GetDetailAvailableMktDepth service returns priceItems as an array of AvailabilityInfo. The table below shows the fields in AvailabilityInfo

Table 71: ArrayOfAvailabilityInfo 0 or more instances of AvailabilityInfo

| Parameter | Type | Description |
|--------------------------|--------|---|
| odds | Double | The price of the bet |
| totalAvailableBackAmount | Double | Total amount available to back at the given odds. This is the |

| Parameter | Type | Description |
|-------------------------|--------|---|
| totalAvailableLayAmount | Double | same as the amounts displayed on the Betfair.com Sports site. Total amount available to lay at the given odds. This is the same as the amounts displayed on the Betfair.com Sports site. |
| totalBspBackAmount | Double | Pre-reconciliation, the total amount of back bets at the requested limit for Limit on Close bets or at 1.01 for Market on Close bets. Post-reconciliation, the total amount of back bets at the reconciled starting price. This field is zero for any odds that are not the starting price. |
| totalBspLayAmount | Double | Pre-reconciliation, the total amount of lay bets at the requested limit for Limit on Close bets or at 1000 for Market on Close bets. Post-reconciliation, the total amount of back bets at the reconciled starting price. This field is zero for any odds that are not the starting price. |

Error Code

The following table describes the error codes returned by GetDetailAvailableMktDepth.

Table 72: GetDetailedAvailMktDepthErrorEnum

| Value | Condition |
|---------------------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_ASIAN_LINE | The asian line specified does not exist |
| INVALID_CURRENCY | The currency specified does not exist or does not match account |
| INVALID_MARKET | The market Id specified does not exist. Check that you have sent your service request to the correct exchange server (the Australian exchange server cannot see markets on the UK exchange server, and vice versa). |
| INVALID_RUNNER | The runner Id specified does not exist |
| MARKET_TYPE_NOT_SUPPORTED | The market Id specified is not supported by the API. |

| Value | Condition |
|------------------|--|
| NO_RESULTS | No results were returned for the request arguments. The market may also be suspended. The minorErrorCode field may contain more information. |
| SUSPENDED_MARKET | The market Id specified refers to a suspended market |

Get Events (global)

The API GetEvents service allows you to navigate through the events hierarchy until you reach details of the betting market for an event that you are interested in.

From API 5.0 onwards, the GetEvents service returns details of line and range markets, where these markets are available for an event.

Requests for the GetEvents service take as input a parameter called eventParentID. The value of this parameter is either:

- the (integer) Id value from one item in an array of eventTypeItems that has been returned by the GetAllEventTypes or GetActiveEventTypes services;
- or an (integer) eventId value from one item in an array of eventItems that has been returned by an earlier GetEvents request.

Use the GetEvents service repeatedly, specifying a different value for eventParentId in each request, until there are no further events to request (this means you have reached the leafnode of the branch of the events tree you have been navigating).

To retrieve full details of a betting market whose details have been returned by the GetEvents service, you need to send a GetMarket request to the exchange server indicated by the market's exchangeId parameter. This GetMarket request must also specify the marketId for the market you are requesting. Both the exchangeId and the marketId are returned by GetEvents. For information about GetMarket, see [Chapter 24](#).

Note:



It is very important that your client software uses the exchangeId parameter in addition to the marketId parameter when requesting market data from the API (using the GetMarket service).

Note:



Failure by a client correctly to identify the exchange server for a particular betting market could have serious consequences. Any PlaceBets requests that a client sends to the wrong exchange server will (for that very reason) not be received by the correct exchange server. Therefore the intended bets will not be placed.

But, even more seriously, because it is possible for a marketId on one exchange server to be duplicated by a marketId on the other, it is also possible for PlaceBets requests sent to the wrong exchange server nevertheless to be processed and matched by that exchange server. (This will occur if the receiving exchange server happens to be hosting a betting market that has the same marketId as has been specified in the PlaceBets request.) In this case, not only will the intended bets not be placed, but other bets that were not intended will be placed instead.

For peace of mind, therefore, as well as using the exchangeId to determine which exchange server to request a betting market from, you might want to code your API client in such a way that, before it places or updates bets, it compares other parameters (in addition to marketId) that are returned by the GetMarket service with their equivalent parameters in the GetEvents response that returned the original details of the market in question.

Table 10-1 on page 32 lists some parameters whose collective agreement in the output of the GetMarket and GetEvents services would enable a client to verify that the market data in both concerned the same market.

For the full list of parameters returned by GetEvents, see the remainder of this chapter. For the full list of parameters returned by GetMarket, see [Chapter 24](#).

Table 73: Parameters to Compare in Case of a Duplication of “marketId” Parameters

| GetEvents Parameters | GetMarket Parameters |
|----------------------|----------------------|
| marketId | marketId |
| marketName | name |
| marketType | marketType |
| startTime | marketTime |

Service Details

SOAP Action: `getEvents`

Input

The following table describes the parameters used for calling the GetEvents service.

Table 74: 1 instance of `GetEventsReq`

| Parameter | Mandatory? | Type | Description |
|---------------|------------|----------------------------------|--|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| eventParentId | Y | Integer | This is either an Id value for a single item (in an array of eventTypeItems returned by GetAllEventTypes or GetActiveEventTypes), or it is an eventId for a single eventItem (in an array of eventItems returned by an earlier GetEvents request). |
| locale | N | String | Specify the language for the reply if you want a different language than the account default. For more information, |

Output

The following table describes the parameters returned from the GetEvents service.

Table 75: 1 Instance of `GetEventsResp`

| Parameter | Type | Description |
|----------------|-----------------------------------|--|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| errorCode | see ... | getEvents specific error code (see enum below) |
| eventItems | ArrayOfBFEvent | Menu nodes beneath the specified parent id |
| eventParentId | Integer | This is either an Event ID or an Event Type ID |
| marketItems | ArrayOfMarketSummary | Summary information on Betfair markets beneath the specified parent id |
| couponLinks | arrayOfCouponLinks | A list of any Betfair coupons that include the market or markets associated with the event returned. (Please note that the data contained in this output parameter is designed for use with a service to be implemented in a future release of the API. Until then the data is redundant.) |
| minorErrorCode | String | Reserved for future use - currently always null |

The `GetEvents` service returns `eventItems` as an array of `BFEvent`. The table below shows the fields in a single `BFEvent`.

Table 76: ArrayOfBFEvent 0 or more instances of `BFEvent`

| Parameter | Type | Description |
|-------------|----------|---|
| eventId | Integer | The ID of the event |
| eventName | String | The name of the event |
| eventTypeid | Integer | The Event Type ID of the event |
| menuLevel | Integer | The depth of the event within the menu |
| orderIndex | Integer | The order in which the event is displayed |
| startTime | DateTime | The start time of the event. This field is only populated for Markets where the market name consists of a time and a name element. For example, 12:30 Bendigo To be placed. |

| Parameter | Type | Description |
|-----------|--------|---------------------------|
| timezone | String | The timezone of the event |

The GetEvents service returns `marketItems` as an array of `MarketSummary`. The table below shows the fields in an instance of `MarketSummary`.

Table 77: ArrayOfMarketSummary 0 or more instances of MarketSummary

| Parameter | Type | Description |
|---------------|---------|--|
| eventTypeId | Integer | The Event Type of the market |
| exchangeId | Integer | The Id of the exchange server that is hosting the betting market for the event you want to bet on. From release 5.0 onwards of the API, the betting market data for an event returned by the getEvents service is available from one of two exchange servers. This means that, to retrieve market data, you need to direct your client software to the correct exchange server for that event. For this reason the getEvents service has a new output parameter called “exchangeId”. The value of this parameter will be ‘1’ for the UK exchange server, and ‘2’ for the Australian Exchange Server. Depending on the value returned, the end-point URLs for requesting betting market data are: https://api.betfair.com/exchange/v3/BFExchangeService , (this is for UK and worldwide sports events, excluding Australian ones), https://api-au.betfair.com/exchange/v3/BFExchangeService , (this is for Australian events). For information about using the getMarket service, see Chapter 24 . To place bets in a market that is operating under an Australian licence you must first transfer funds into your Australian wallet. For more information, see Chapter 54 . |
| eventParentId | Integer | This is either an Event ID or an Event Type ID |
| marketId | Integer | The ID of the market |

| Parameter | Type | Description |
|-------------------|--------------------------------|--|
| marketName | String | The name of the market |
| marketType | MarketTypeEnum | The market type (e.g. Asian Handicap, Odds). For more information, see Chapter 61 . |
| marketTypeVariant | | The market variant for Asian Handicap markets |
| menuLevel | Integer | The depth of the market within the menu |
| orderIndex | Integer | The order in which the event is displayed |
| startTime | Date/Time | The start time of the market - to be used for display purposes (the value can be NULL). |
| timezone | String | The timezone of the event |
| venue | String | The location/venue of the event |
| betDelay | Integer | The betting delay that applies to this market when placed in-play. This is non-zero when the market is in-play. |
| numberOfWinners | Integer | The number of possible winning selections in the betting market. This parameter helps you to determine whether the market is a win or place market. If the value is '1', it is a win market; if it is '2', the market is a place market. |

The GetEvents service returns `CouponLinks` as an array of instances of `CouponLink`. The table below shows the fields in a single instance of `CouponLink`. A `CouponLink` identifies any Betfair coupons that include the market or markets associated with an event. Please note that the data it contains is designed for use with a service to be implemented in a future release of the API (that will enable you to request particular coupons from the API). Until then the data is redundant.

Table 78: ArrayOfCouponLinks 0 or more instances of CouponLink

| Parameter | Type | Description |
|------------|---------|---|
| couponId | Integer | The unique numerical identifier of a coupon |
| couponName | String | The name of the coupon |

Error Code

The following table describes the error codes returned by GetEvents.

Table 79: GetEventsErrorEnum

| Value | Condition |
|--------------------------------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_EVENT_ID | The parent id is either not valid or the parent does not have any event children |
| INVALID_LOCALE_DEFAULTING_TO_ENGLISH | The locale string was not recognized. Returned results are in English |
| NO_RESULTS | No data available to return |

**Note:**

This replaces the old API services GetSportsForEvent and GetEventsForEventTypes.

The market field Display Start Date is only populated for Markets where the market name consists of a 'time' and a 'name' element (e.g. 12:30 Bendigo To be placed), as this market name is actually stored separately as the 'time' and 'name' element. Note that events or menu nodes are now kept in a separate object from actual markets. This is a change from API 3.0 where markets and event nodes had the same record.

Get In-Play Markets (exchange)

The API GetInPlayMarkets service allows you to retrieve the markets that will be turned in-play in the next 24 hours. You can use this service to duplicate the In-Play Today listing of the Betfair.com web site. The service does not return markets that are closed or that were turned in-play prior to the time the service was called.

**Note:**

You should not use this service to poll the in-play status of a market. If you want to detect the moment a market turns in-play, use the GetMarketPricesCompressed service.

The return format is similar to the GetAllMarkets service.

Service Details

SOAP Action: getInPlayMarkets

Known Issues

All BF Tennis, Golf, Cricket and Rugby Union markets are showing a prefix of 'Group A/B/C/D' in the menuPath, while there is no such valid component.

Input

The following table describes the parameters used for calling the GetInPlayMarkets service.

Table 80: 1 instance of GetInPlayMarketsReq

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token. <i>See "API Request/Response Header" on page 21.</i> |

| Parameter | Mandatory? | Type | Description |
|-----------|------------|--------|--|
| locale | N | String | The locale to use when returning results. If not specified, the default locale for the user's account is used. |

Output

The following table describes the parameters returned from the GetInPlayMarkets service.

Table 81: 1 Instance of GetInPlayMarketsResp

| Parameter | Type | Description |
|----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token. <i>See “API Request/Response Header” on page 21.</i> |
| errorCode | see ... | Service specific error code |
| marketData | String | Delimited field containing market data (see below) |
| minorErrorCode | String | Reserved for future use - currently always null |

The GetInPlayMarkets service returns a string representing the market data. The table below shows the fields in marketData. Each market is delimited by a : (colon). Fields within a market are delimited by a ~ (tilde).

Table 82: marketData

| Field No. | Type | Name | Notes |
|-----------|---------|-------------|--|
| 1 | Integer | Market ID | The Market Id of the market. Note: This is unique to an exchange, but is not guaranteed to be unique between exchanges. Always check the ExchangeID of the market. |
| 2 | String | Market name | The displayed name of the market. The name is returned in the account's default locale or in the locale specified in the request. |
| 3 | String | Market Type | The type of the market. |

| Field No. | Type | Name | Notes |
|-----------|----------|-------------------|---|
| 4 | String | Market Status | The current status of the market (i.e., active or suspended). |
| 5 | DateTime | Event Date | The date and time the event starts (in milliseconds since January 1 1970 00:00:00 GMT) |
| 6 | String | Menu Path | The textual menu path for the market, not including the Market name at the end. The path is returned in the account's default locale or in the locale specified in the request. |
| 7 | String | Event Hierarchy | The full hierarchy of Event IDs leading to the specified market, including the Market Id. |
| 8 | String | Bet Delay | The current bet delay. This will be non-zero when the market is in-play. |
| 9 | Integer | Exchange Id | The exchange Id for the market. 1 for the UK/Worldwide exchange, 2 for the Australian Exchange. |
| 10 | String | ISO3 Country Code | The Country Code for the event. An empty field indicates an international market that takes place in multiple countries. |
| 11 | DateTime | Last Refresh | The time (in milliseconds since January 1 1970 00:00:00 GMT) since the cached market data was last refreshed from the exchange database. The API caches market information for 5 minutes. |

| Field No. | Type | Name | Notes |
|-----------|---------|----------------------|--|
| 12 | Integer | Number of Runners | The number of runners in the market |
| 13 | Integer | Number of Winners | The number of possible winners in the market |
| 14 | Double | Total Amount Matched | The total amount of money in GBP that have been matched on the specified market. |
| 15 | Boolean | BSP Market | If Y, indicates that the market supports Betfair Starting Price bets. |
| 16 | Boolean | Turning In Play | If Y, indicates that the market is scheduled to be turned in-play |

Example marketData output

```
:Soccer~20158165~Match Odds~O~ACTIVE~1164223800000~\Soccer
\Scottish Soccer\Bells League Div 1\Fixtures 22 November \Partick v
Clyde~/1/2695886/610072/10551708/10551709/20158165~0~1~GBR~1164192924479~3~1~8737.44~N~Y:
```

Note:

The response may contain an ‘escape’ character, a backslash (\), to indicate that the subsequent character should be interpreted literally and not as a delimiter. For example, for markets with event times in the name, the response would look like: ...15\;35...

Note:

To recreate a textual menu that matches the numeric hierarchy, you must append the Market name returned (i.e., Partick v Clyde).

Error Code

The following table describes the error codes returned by GetInPlayMarkets.

Table 83: GetInPlayMarketsErrorEnum

| Value | Condition |
|----------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_LOCALE | The locale specified does not exist. See “Locale Specification” on page 211 |

Get Market (exchange)

The API GetMarket service allows the customer to input a Market ID and retrieve all static market data for the market requested. # To get a Market ID for the betting market associated with an event you are interested in, use the GetEvents command.

Service Details

SOAP Action: `getMarket`

Known Issues

All BF Tennis, Golf, Cricket and Rugby Union markets are showing a prefix of 'Group A/B/C/D' in the menuPath, while there is no such valid component.

The market's Parent Event name in the menu path returned in the response to getMarket and getAllMarkets is sometimes truncated.

Input

The following table describes the parameters used for calling the GetMarket service.

Table 84: 1 instance of `GetMarketReq`

| Parameter | Mandatory? | Type | Description |
|--------------------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| marketId | Y | Integer | The market ID. |
| includeCouponLinks | N | Boolean | If you set this parameter to true, the service response contains a list of any coupons that include the market you have requested. If you set the parameter to false, no coupon data is returned. |
| locale | N | String | Specify the language for the reply if you want a different language than the account default. For more information, |

Output

The following table describes the parameters returned from the GetMarket service.

Table 85: 1 Instance of `GetMarketResp`

| Parameter | Type | Description |
|----------------|------------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| errorCode | GetMarketErrorEnum | getMarket specific error code (see enum below) |
| market | Market | The market data |
| minorErrorCode | String | Reserved for future use - currently always nulls |

The GetMarket service returns Market. The table below shows the fields in Market

Table 86: Market

| Parameter | Type | Description |
|-----------------|--------------------|--|
| bspMarket | Boolean | If true, indicates that the market supports Betfair Starting Price bets. |
| countryISO3 | String | The ISO3 country code where the event is taking place |
| couponLinks | arrayOfCouponLinks | A list of the coupons that include this 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. |
| eventHierarchy | ArrayOfEventId | The eventHierarchy is a history of the eventIds you have specified in the series of getEvents requests that brought you to the market that is now the subject of your getMarket request. The eventHierarchy array returned by the getMarket request is empty in output received from the Australian exchange server. |
| eventTypeid | Integer | The event type id of the market. |
| interval | Integer | This parameter states the interval between the prices you can bet at in the market. (The term "price increment" is often used interchangeably |

| Parameter | Type | Description |
|--------------------------|---------------------------------------|--|
| lastRefresh | Long | with “interval”; it has the same meaning.) This field is deprecated and it should not be relied on for determining the order or age of responses. |
| licenceId | Integer | The licence under which the betting market is operating. The value will be ‘1’ for the UK licence and ‘2’ for the Australian licence. |
| marketBaseRate | Float | This will indicate the commission price set for each market. |
| marketDescription | String | The text associated with the market containing market-specific information (not including the market’s start date and time even if it has one) and rules. |
| marketDescriptionHasDate | Boolean | If this parameter is set to true, then the market has a start date and time and your client application should append it to the marketDescription (see above). |
| marketDisplayTime | DateTime | The time used to refer to the market – normally relevant for horse race – e.g. the 3:30 Haydock. |
| marketId | Integer | Id of the requested market |
| marketStatus | MarketStatusEnum | Current status of the market. For more information, see Chapter 61 . |
| marketSuspendTime | DateTime | The time the market will next be suspended |
| marketTime | DateTime | The expected start time of the market |
| marketType | MarketTypeEnum | The type of market (Odds or Asian Handicap). For more information, see Chapter 61 . |
| marketTypeVariant | MarketTypeVariantEnum | The Market variant for Asian Handicap markets. For more information, see Chapter 61 . |
| maxUnitValue | Integer | The maximum unit value (for example, the maximum number of goals) that bets |

| Parameter | Type | Description |
|-------------------|----------------|--|
| | | can be matched against in this market. |
| menuPath | String | The detailed path through the Betfair menu to reach this market |
| minUnitValue | Integer | The minimum unit value (for example, the minimum number of goals) that bets can be matched against in this market. |
| name | String | The name of the market |
| numberOfWinners | Integer | How many winners there are in this market (e.g. 1 for win markets, but 2,3 or 4 for place markets) |
| parentEventId | Integer | Id of the parent Event node (currently not populated) |
| runners | ArrayOfRunners | Details of the runners in the market. Is empty for settled markets. |
| runnersMayBeAdded | Boolean | True if and only if new runners may be subsequently added to the market |
| timezone | String | The timezone where the market is taking place |
| unit | Integer | This parameter is for line and range betting markets. It tells you whether the unit totals available to bet on represent runs, goals, points, shots or possibly some other type of unit. |

The `GetMarket` service returns `couponLinks` as an `arrayOfCouponLinks`. The table below shows the fields in an instance of `CouponLink`. A `CouponLink` identifies any Betfair coupons that include the market or markets associated with an event. Please note that the data it contains is designed for use with a service to be implemented in a future release of the API (that will enable you to request particular coupons from the API). Until then the data is redundant.

Table 87: ArrayOfCouponLinks 0 or more instances of CouponLinks

| Parameter | Type | Description |
|------------|---------|---|
| couponId | Integer | The unique numerical identifier of a coupon |
| couponName | String | The name of the coupon |

The `GetMarket` service returns `runners` as an array. The table below shows the fields in an instance of `Runners`

Table 88: ArrayOfRunners 0 or more instances of Runners

| Parameter | Type | Description |
|-------------|---------|---|
| asianLineId | Integer | Id of the selection (this will be the same for the same selection across markets) |
| handicap | Double | Handicap of the market (applicable to Asian handicap markets) |
| name | String | Runner name |
| selectionId | Integer | Runner Id |

Error Code

The following table describes the error codes returned by GetMarket.

Table 89: GetMarketErrorEnum

| Value | Condition |
|--------------------------------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_LOCALE_DEFAULTING_TO_ENGLISH | The locale string was not recognized. Returned results are in English |
| INVALID_MARKET | Invalid Market ID supplied. Make sure you have sent the request to the correct exchange server. Check that you have sent your service request to the correct exchange server (the Australian exchange server cannot see markets on the UK exchange server, and vice versa). |
| MARKET_TYPE_NOT_SUPPORTED | The MarketID supplied identifies a market of a type that is not supported by the API. |

Get Market Info (exchange)

The API GetMarketInfo service allows you to input a Market ID and retrieve market data for the market requested. To get a Market ID for the betting market associated with an event you are interested in, use the GetEvents command. This is a lite service to compliment the GetMarket service.

Service Details

SOAP Action: getMarketInfo

Input

The following table describes the parameters used for calling the GetMarketInfo service.

Table 90: 1 instance of `GetMarketInfoReq`

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|--|
| header | Y | APIRequestHeader | The API Request Header that contains the session token (see API Request/Response Header). |
| marketId | Y | Integer | The market ID. |

Output

The following table describes the parameters returned from the `GetMarketInfo` service.

Table 91: 1 Instance of `GetMarketInfoResp`

| Parameter | Type | Description |
|----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header (see API Request/Response Header). |
| errorCode | see ... | getMarketInfo specific error code (see enum below) |
| marketLite | see ... | The market data |
| minorErrorCode | String | Reserved for future use - currently always nulls |

The `GetMarketInfo` service returns an array called `MarketLite`. The table below shows the fields in `MarketLite`

Table 92: `MarketLite`

| Parameter | Type | Description |
|-------------------|----------------------------------|--|
| delay | Integer | The number of seconds delay between submission and a bet actually being placed. This is greater than 0 if and only if the market is in-play. |
| marketStatus | MarketStatusEnum | Current status of the market. For more information, see Chapter 61 . |
| marketSuspendTime | DateTime | The time the market will next be suspended |
| marketTime | DateTime | The expected start time of the market |
| numberOfRunners | Integer | The number of runners in the market |
| openForBspBetting | Boolean | If true, indicates that the market supports Betfair Starting Price bets. |

Error Code

The following table describes the error codes returned by GetMarketInfo.

Table 93: GetMarketErrorEnum

| Value | Condition |
|--------------------------------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_LOCALE_DEFAULTING_TO_ENGLISH | The locale string was not recognized. Returned results are in English |
| INVALID_MARKET | Invalid Market ID supplied. Make sure you have sent the request to the correct exchange server. Check that you have sent your service request to the correct exchange server (the Australian exchange server cannot see markets on the UK exchange server, and vice versa). |
| MARKET_TYPE_NOT_SUPPORTED | The MarketID supplied identifies a market of a type that is not supported by the API. |

Get Market Prices (exchange)

The API GetMarketPrices service allows you to retrieve dynamic market data for a given Market ID.

Service Details

SOAP Action: getMarketPrices

Known Issues

You may see additional prices displayed on the Betfair.com website than are returned by the API. For more information see: https://bdp.betfair.com/index.php?option=com_content&task=view&id=143&Itemid=62

Input

The following table describes the parameters used for calling the GetMarketPrices service.

Table 94: 1 instance of GetMarketPricesReq

| Parameter | Mandatory? | Type | Description |
|--------------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| currencyCode | N | String | Three letter ISO 4217 code. If not supplied, users currency is used |

| Parameter | Mandatory? | Type | Description |
|-----------|------------|---------|---------------------------|
| marketId | Y | Integer | ID of the required market |

Output

The following table describes the parameters returned from the GetMarketPrices service.

Table 95: 1 Instance of GetMarketPricesResp

| Parameter | Type | Description |
|----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| errorCode | see ... | getMarketPrices specific error code (see enum below) |
| marketPrices | see ... | Object containing the prices of the runners |
| minorErrorCode | String | Reserved for future use – currently always null |

The table below shows the fields in `MarketPrices`

Table 96: MarketPrices

| Parameter | Type | Description |
|-----------------|---------|---|
| bspMarket | Boolean | If true, indicates that the market supports Betfair Starting Price bets. |
| currencyCode | String | Three letter ISO 4217 code |
| delay | Integer | The number of seconds delay between submission and a bet actually getting placed. This is greater than 0 if and only if the market is in-play. |
| 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. |
| lastRefresh | Long | This field is deprecated and it should not be relied on for determining the order or age of responses. |
| marketBaseRate | Float | This will indicate the commission price set for each market. |
| marketId | Integer | ID of the market |

| Parameter | Type | Description |
|-----------------|----------------------------------|--|
| marketInfo | String | The text associated with the market containing market specific information and rules. This part of the text contains dynamic information such as non-runners |
| marketStatus | MarketStatusEnum | Status of the market - please see Chapter 61 . |
| numberOfWinners | Integer | How many winners there are in this market (e.g. 1 for win markets, but 2,3 or 4 for place markets) |
| removedRunners | String | A list of non-runners in the market. The removedRunners string has the following format: "RACE CARD NUMBER1, HORSE NAME1, GMT TIME OF REMOVAL1, REDUCTION FACTOR %1; RACE CARD NUMBER2, HORSE NAME2, GMT TIME OF REMOVAL2, REDUCTION FACTOR %2; ". |
| runnerPrices | see ... | Empty if market is not active |

The GetMarketPrices service returns `runnerPrices` as an array of `RunnesPrices`. The table below shows the fields in `RunnerPrices`

Table 97: ArrayOfRunnerPrices

| Parameter | Type | Description |
|------------------|-------------------------|--|
| actualBSP | Double | The actual Betfair Starting Price. Before the starting price is determined, this field contains zero. |
| asianLineId | Integer | Id of the selection (this will be the same for the same selection across markets) |
| bestPricesToBack | see ... | Best available back prices |
| bestPricesToLay | see ... | Best available lay prices |
| farBSP | 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. |

| Parameter | Type | Description |
|--------------------|---------|---|
| handicap | Double | Handicap of the market (applicable to Asian handicap markets) |
| lastPriceMatched | Double | The last price at which this selection was matched |
| nearBSP | 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. |
| reductionFactor | Double | Reduction in the odds that applies in case this runner does not participate |
| selectionId | Integer | Id of the selection (this will be the same for the same selection across markets) |
| sortOrder | Integer | The order in which the items are displayed on Betfair |
| totalAmountMatched | Double | The total amount matched on this selection (regardless of price) |
| vacant | Boolean | Used to indicate a Vacant Trap for withdrawn runners in Greyhound markets |

The GetMarketPrices service returns `bestPricesToBack` and `bestPricesToLay` as an array of Price. The table below shows the fields in ArrayOfPrice

Table 98: ArrayOfPrice

| Parameter | Type | Description |
|-----------------|--------------------|---|
| amountAvailable | Double | Amount available at the odds specified. |
| betType | <i>BetTypeEnum</i> | Bet type data. For more information, see Chapter 61 . |
| depth | Integer | The order, from best to worst, of the price ('1' is best available) |
| price | Double | Odds |



Note:

Type Double may contain numbers, INF, -INF, and NaN.

Error Code

The following table describes the error codes returned by GetMarketPrices.

Table 99: GetMarketPricesErrorEnum

| Value | Condition |
|---------------------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_CURRENCY | Currency code not a valid 3 letter ISO 4217 currency abbreviation |
| INVALID_MARKET | Market Id is not valid. Check that you have sent your service request to the correct exchange server (the Australian exchange server cannot see markets on the UK exchange server, and vice versa). |
| MARKET_TYPE_NOT_SUPPORTED | The MarketID supplied refers to a market that is not supported by the API. Currently, this includes Line and Range markets. |

Get Market Prices Compressed (exchange)

The API GetMarketPricesCompressed service allows you to retrieve dynamic market data for a given Market ID in a compressed format. This service returns the same information as the Get Market Prices service but returns it in a ~ (tilde) delimited String.

Service Details

SOAP Action: `getMarketPricesCompressed`

Known Issues

You may see additional prices displayed on the Betfair.com website than are returned by the API. For more information see: https://bdp.betfair.com/index.php?option=com_content&task=view&id=143&Itemid=62

Input

The following table describes the parameters used for calling the GetMarketPricesCompressed service.

Table 100: 1 instance of GetMarketPricesCompressedReq

| Parameter | Mandatory? | Type | Description |
|--------------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| currencyCode | N | String | The three letter ISO 4217 code. If not supplied, user's currency is used |
| marketId | Y | Integer | ID of the desired market |

Output

The following table describes the parameters returned from the GetMarketPricesCompressed service.

Table 101: 1 Instance of GetMarketPricesCompressedResp

| Parameter | Type | Description |
|----------------|---------------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| errorCode | MarketPricesErrorEnum | Specific error code (see enum below) |
| marketPrices | String | Delimited field containing market price records (see below) |
| minorErrorCode | String | Reserved for future use - currently always null |

The GetMarketPricesCompressed service returns a string representing the compressed market prices. The table below shows the fields in

Table 102: marketPrices

| Field No. | Type | Name | Notes |
|-----------|------------------|--------------------|--|
| 1 | Integer | Market ID | |
| 2 | String | Currency | Three letter ISO currency code in ISO 4217 format |
| 3 | MarketStatusEnum | Market Status | Status of the market |
| 4 | Integer | In Play Delay | The number of seconds delay between submission and a bet actually getting placed. This is greater than 0 if and only if the market is in-play. |
| 5 | Integer | Number of Winners | How many winners there are in this market (e.g. 1 for win markets, but 2,3 or 4 for place markets) |
| 6 | String | Market Information | May be null |
| 7 | Boolean | Discount Allowed? | Does user commission discount count against Market Base Rate? |

| Field No. | Type | Name | Notes |
|---|---------|------------------------------|--|
| 8 | String | Market Base Rate | Base rate of commission on market |
| 9 | Long | Refresh time in milliseconds | This field is deprecated and it should not be relied on for determining the order or age of responses. |
| Removed runner information fields delimited by , (commas) then delimited by a ; (semicolon), 0 or more iterations | | | |
| 1 | String | Selection Name | The name of the removed runner |
| 2 | Time | Removed Time | The time the runner was removed in the format "H.mm" where H is the hour of the day (0 - 23) and mm is the minute of the hour (00 - 59). |
| 3 | String | Reduction Factor | The reduction factor applied to the remaining runners |
| Betfair Starting Price Information | | | |
| 1 | Boolean | BSP Market | Y if the market allows Betfair SP bets, otherwise N. |
| Runner information fields then delimited by a : (colon), 0 or more iterations | | | |
| 1 | Integer | Selection ID | Id of the selection (this will be the same for the same selection across markets) |

| Field No. | Type | Name | Notes |
|-----------|---------|----------------------|---|
| 2 | Integer | Order Index | The order in which the items are displayed on Betfair |
| 3 | Double | Total Amount Matched | The total amount matched on this selection (regardless of price) |
| 4 | Double | Last Price Matched | The last price at which a selection was matched |
| 5 | Double | Handicap | Handicap of the market (applicable to Asian handicap markets) |
| 6 | Double | Reduction Factor | Reduction in the odds that applies in case this runner does not participate |
| 7 | Boolean | Vacant | Used to indicate a Vacant Trap for withdrawn runners in Greyhound markets |
| 8 | Double | Far SP Price | A prediction of the eventual starting price. 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 exchange at SP. |
| 9 | Double | Near SP Price | A prediction of the eventual starting price. 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. |
| 10 | Double | Actual SP Price | The actual Betfair Starting Price. Before the starting price is determined, this field contains zero. |

| Field No. | Type | Name | Notes |
|--|---------|------------------|--|
| Back Prices then delimited by a (pipe), 0 or more iterations | | | |
| 1 | Double | Price | Odds |
| 2 | Double | Amount Available | Amount available at the odds specified |
| 3 | String | | For Back prices, this will always be L as the prices available to back are made up of the existing Lay bets already on the Exchange. |
| 4 | Integer | Depth | The order, from best to worst, of the price ('1' is best available) |
| Lay Prices then delimited by a (pipe), 0 or more iterations | | | |
| 1 | Double | Price | Odds |
| 2 | Double | Amount Available | Amount available at the odds specified |
| 3 | String | | For Lay prices, this will always be B as the prices available to lay are made up of the existing Back bets already on the Exchange. |
| 4 | Integer | Depth | The order, from best to worst, of the price ('1' is best available) |

Example marketPrices output

```
5082333~GBP~ACTIVE~0~1~NR\ : (RSA) <br>8. Fan
Mail (0%,11\ :07), 6(2.5%,11\ :08)~true~5.0~1162835723938~6.
Earlswood,9.08,2.5;8.
Fan
Mail,9.07,2.4;Y:1058616~0~6.04~8.4~~11.9~false|| :670160~1~6.18~17.5~~4.2~false|| :
1.01~5.0~B~1~:1414415~4~8.2~6.2~~16.0~false|| :575636~5~5.54~11.5~~8.6~false|| :1433
0.9~false|| :1433460~13~0.0~~~0.9~false||
```

**Note:**

The response may contain an 'escape' character, a backslash (\), to indicate that the subsequent character should be interpreted literally and not as a delimiter. For example, in the Market Info field, you may see the following: NR\; (AEST).

Type Double may contain numbers, INF, -INF, and NaN.

Error Code

The following table describes the error codes returned by GetMarketPricesCompressed.

Table 103: MarketPricesErrorEnum

| Value | Condition |
|---------------------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_CURRENCY | Currency code not a valid 3 letter ISO 4217 currency abbreviation |
| INVALID_MARKET | Market ID is not a valid market id. Check that you have sent your service request to the correct exchange server (the Australian exchange server cannot see markets on the UK exchange server, and vice versa). |
| MARKET_TYPE_NOT_SUPPORTED | The MarketID supplied refers to a market that is not supported by the API. |

Get Matched and Unmatched Bets (exchange)

The API GetMUBets service allows you to retrieve information about all your matched and unmatched bets on a particular exchange server.



Note:

You should be aware that voided bets are not returned by getMUBets. Your application should track the number of matched and unmatched bets against the number of bets returned by getMUBets in order to detect a voided bet.

Service Details

SOAP Action: getMUBets

Known Issues

Bet prices of 1.01 or 1000 are returned for BSP Market on Close bets instead of 0.0. Ignore the contents of the price field when betCategoryType is M and the bet status is U.

The avgPrice is not returned for a matched BSP Market on Close bet when the market is suspended. Ignore the contents of s avgPrice when market status is 'suspended' and bet category type is 'market on close'. The avgPrice field should have the same value as the priceMatched field.

Input

The following table describes the parameters used for calling the GetMUBets service.

Table 104: 1 instance of `GetMUBetsReq`

| Parameter | Mandatory? | Type | Description |
|-------------------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| betIds | N | ArrayOfBetId | Specifies the betId of each bet you want returned. The maximum number of bets you can include in the array is 200. If you include marketId in the request and marketId contains a non-zero value, then betIds is ignored. If you specify a betId, then you must specify MU for betStatus. |
| betStatus | Y | BetStatusEnum | The status of the bets to return (matched, unmatched, or both) - please see Betfair Simple Data Types . If you specify a betId, then you must specify MU. |
| excludeLastSecond | N | Boolean | If true, the API excludes bets placed or matched that occurred less than one second before the GetMUBets call. Set this to true if you want to ensure that the response does not include bets that may have changed state between the time you sent the request and before the response was generated. If false, all bets are returned. Therefore, you may receive a response that indicates an unmatched bet that has actually been matched during the |

| Parameter | Mandatory? | Type | Description |
|--------------|------------|------------------------|---|
| marketId | N | Integer | time taken for the API to respond. If marketId is present and non-zero, then bets placed on the specified market are returned and any bet ids specified in betIds are ignored. |
| matchedSince | N | dateTime | Specifies a date and time to start from for the list of returned bets. (The dateTime datatype is pre-defined in XML. For information about its format, check the following URL: http://www.w3.org/TR/xmlschema-2/#dateTime .) If you use the matchedSince parameter and you have specified a betStatus of MU, the bets returned will ignore any limit you set (using recordCount) for the number of records to be returned. Specifying a betStatus of MU causes the API to return your unmatched bets along with the matched ones. |
| orderBy | Y | <i>BetsOrderByEnum</i> | The order of returned results. If Ordering is not NONE, please see <i>Betfair Simple Data Types</i> . Valid orders are BET_ID, PLACED_DATE, and MATCHED_DATE. |
| recordCount | Y | Integer | Maximum number of records to return. The maximum number allowed is 200. |

| Parameter | Mandatory? | Type | Description |
|-------------|------------|-------------------------------|--|
| sortOrder | Y | SortOrderEnum | Whether the results are in ascending or descending order |
| startRecord | Y | Integer | The first record number to return (supports paging). Record numbering starts from 0. For example, to retrieve the third record and higher, set startRecord to 2. |

The following table shows the ordering for each Bet status and bet ordering requested. Bets are sorted first by the selected ordering and then sorted by either bet id or internal transaction id.

Table 105: Ordering Combinations by Bet status

| Order By | Bet Status | | | | | |
|--------------------|----------------|--------------------|----------------|--------------------|---------------------|----------------|
| Matched | Unmatch | Matched/Unmatched | | | | |
| First sorted by... | then by... | First sorted by... | then by... | First sorted by... | then by... | |
| NONE | Transaction ID | N/A | Transaction ID | N/A | Transaction ID | N/A |
| BET_ID | Bet ID | Transaction ID | Bet ID | Transaction ID | Bet ID | Transaction ID |
| PLACED_DATE | Placed Date | Bet Id | Placed Date | Bet Id | Placed Date | Bet Id |
| MATCHED_DATE | Matched Date | Bet Id | N/A | N/A | Matched Date Bet Id | Bet Id |

Output

The following table describes the parameters returned from the GetMUBets service.

Table 106: 1 Instance of `GetMUBetsResp`

| Parameter | Type | Description |
|-----------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| bets | ArrayOfMUBet | The Bet information |
| errorCode | String | Specific error code (see enum below) |

| Parameter | Type | Description |
|------------------|---------|---|
| minorErrorCode | String | Reserved for future use - currently always null |
| totalRecordCount | Integer | Total number of records available |

The GetMUBets service returns `bets` as an array of MUBet. The table below shows the fields in MUBet

Table 107: ArrayOfMUBet 0 or more instances of MUBet

| Parameter | Type | Description |
|--------------------|--|--|
| asianLineId | Integer | Id of the specific Asian handicap line |
| betCategoryType | BetCategoryTypeEnum | Exchange, Market on Close SP bet, or Limit on Close SP bet. |
| betId | Long | Unique identifier generated for every bet placement |
| betPersistenceType | BetPersistenceTypeEnum | What happens to the bet when the market turns in-play. |
| betStatus | BetStatusEnum | Status of the bet. Note: getMUBets returns only either M or U as the betStatus for an individual bet or portion of a bet. |
| betType | BetTypeEnum | Back or lay - please see Betfair Simple Data Types . |
| bspLiability | Double | The maximum liability for the BSP bet. |
| executedBy | String | Used for internal Betfair applications. |
| handicap | Double | Used for internal Betfair applications. |
| marketId | Integer | Id of the market |
| matchedDate | dateTime | Date and time that the bet was matched (null if not applicable) |
| placedDate | dateTime | Date and time of bet placement |
| price | Double | For matched bets (or matched portion of a bet), the price that was matched. For unmatched bets (or unmatched portions of a bet), the originally submitted price. |
| selectionId | Integer | Id of the selection (this will be the same for the same selection across markets) |
| size | Double | If betStatus is M, size is the amount matched. If |

| Parameter | Type | Description |
|---------------|------|--|
| transactionId | Long | betStatus is U, size is the amount unmatched. The transaction Id for this portion of the bet. |

Error Codes

The following table describes the error codes returned by GetMUBets.

Table 108: GetMUBetsErrorEnum

| Value | Condition |
|-----------------------------|--|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_BET_STATUS | Invalid bet status. The bet status must be M, U, or MU. If a betId is specified, then bet status must be MU. |
| INVALID_MARKET_ID | Market ID is negative or does not exist. Check that you have sent your service request to the correct exchange server (the Australian exchange server cannot see markets on the UK exchange server, and vice versa). |
| INVALID_ORDER_BY_FOR_STATUS | Ordering is not NONE and: <ol style="list-style-type: none"> 1. Bet Status is M (matched) and Ordering is neither MATCHED_DATE or PLACED_DATE 2. Bet Status is U (unmatched) and ordering isn't PLACED_DATE Bet Status is MU (matched and unmatched) and Ordering is neither MATCHED_DATE or PLACED_DATE |
| INVALID_RECORD_COUNT | Record Count is negative. |
| INVALID_START_RECORD | Start record is not supplied or is not greater than or equal to 1. |
| NO_RESULT | There were no matched or unmatched bets found |
| TOO_MANY_BETS_REQUESTED | You submitted an array of betIds for more than 200 bets. |

Get Market Profit And Loss (exchange)

The API GetMarketProfitAndLoss service allows you to Retrieve Profit and Loss information for the user account in a given market.

The limitations for the service in the initial release are:

- Profit and loss for single and multi-winner odds markets is implemented however it won't calculate `worstCaseIfWin` nor `futureIfWin`.
- The calculation for AH markets will include `worstCaseIfWin` but not `futureIfWin`.

Service Details

SOAP Action: `getMarketProfitAndLoss`

Input

The following table describes the parameters used for calling the `GetMarketProfitAndLoss` service.

Table 109: 1 instance of `GetMarketProfitAndLossReq`

| Parameter | Mandatory? | Type | Description |
|---------------------------------|------------|----------------------------------|--|
| <code>header</code> | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| <code>locale</code> | N | String | Specify the language for the reply if you want a different language than the account default. For more information, |
| <code>includeSettledBets</code> | N | Boolean | If Y, then the P&L calculation for each runner includes any profit and loss from any bets on runners that have already been settled. The default is N, which matches the default on Betfair.com. |
| <code>includeBspBets</code> | Y | Boolean | If yes, BSP bets are returned as part of the P&L |
| <code>marketID</code> | Y | Integer | The market ID for which the profit and loss for the user is to be returned |
| <code>netOfCommission</code> | N | Boolean | If yes, return P&L net of users current commission rate for this market including any special tariffs, default is FALSE. |

Output

The following table describes the parameters returned from the `GetMarketProfitAndLoss` service.

Table 110: 1 Instance of `GetMarketProfitAndLossResp`

| Parameter | Type | Description |
|---------------------|-----------------------------------|--|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| annotations | ArrayOfProfitAndLoss | P&L data |
| commissionApplied | Double | The commission rate applied to the P&L numbers. If 0, this implies that no commission has been deducted from the P&L returned. |
| currencyCode | String | The currency for all amounts returned (this is the account currency) |
| errorCode | see ... | Specific error code (see enum below) |
| includesSettledBets | Boolean | TRUE if and only if any settled bets are included in the P&L position (regardless of the Settled Bets input) |
| includesBspBets | Boolean | TRUE if an only if any BSP bets are included in the P&L position. |
| marketId | Integer | The market ID for which the profit and loss for the user is to be returned |
| marketName | String | Name of the market |
| marketStatus | MarketStatusEnum | Status of the market - please see Chapter 61 . |
| minorErrorCode | String | Reserved for future use - currently always null |
| unit | String | For asian handicap markets, the unit the handicaps are expressed in. "N/A" for odds markets. |

The `GetMarketProfitAndLoss` service returns `ProfitAndLoss` as an array.

Single Winner Odds Markets

The table below shows the fields in `ProfitAndLoss` This structure is used for a single winner odds market.

Table 111: ArrayOfProfitAndLoss 0 or more instances of ProfitAndLoss

| Parameter | Type | Description |
|-------------|--------|-----------------|
| futureIfWin | Double | (Null) NOT USED |

| Parameter | Type | Description |
|----------------|--------|--|
| ifWin | Double | If this selection is the winner, this is the profit and loss for the market (based on matched bets only) |
| selectionID | Int | The selection ID |
| selectionName | String | The selection Name |
| worstCaseIfWin | Double | (null) NOT USED |

Asian Handicap Markets

Each instance associates an outcome in a handicap market with a profit/loss value. The outcome is specified as an interval with the 'from' and 'to' values either expressing units or unit differences. Where the values express a unit difference, it is from the perspective of the team specified in fields selection id/name. In the latter case, a positive value is a win, a negative value is a loss and a zero value is a draw. The units are expressed in the units of the handicap (goals, points, runs etc.) Infinite intervals express "more than" and "less than" outcomes, and positive/negative infinite values are used.

The table below shows the fields in `HandicapLine`. This structure is used for an asian handicap market

Table 112: HandicapLine

| Parameter | Type | Description |
|----------------|--------|--|
| futureIfWin | Double | (Null) NOT USED |
| ifWin | Double | If this selection is the winner, this is the profit and loss for the market (based on matched bets only) |
| selectionID | Int | The selection ID |
| selectionName | String | The selection Name |
| worstCaseIfWin | Double | (null) NOT USED |
| from | Double | The from value; includes -infinity |
| to | Double | The from value; includes +infinity |

Multi-Winner Odds Market

The table below shows the fields in `MultiWinnerOddsLine`. This structure is used for a multi-winner odds market

Table 113: MultiWinnerOddsLine

| Parameter | Type | Description |
|-------------|--------|--|
| futureIfWin | Double | (Null) NOT USED |
| ifWin | Double | If this selection is the winner, this is the profit and loss for the market (based on matched bets only) |
| selectionID | Int | The selection ID |

| Parameter | Type | Description |
|----------------|--------|---|
| selectionName | String | The selection Name |
| worstCaseIfWin | Double | (null) NOT USED |
| ifLoss | Double | If this selection is the loser, this is the profit and loss for the market (based on matched bets only) |

Error Codes

The following table describes the error codes returned by GetMarketProfitAndLoss.

Table 114: GetMarketProfitAndLossErrorEnum

| Value | Condition |
|-------------------------|--|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_MARKET | The specified market is invalid. Check that you have sent your service request to the correct exchange server (the Australian exchange server cannot see markets on the UK exchange server, and vice versa). |
| MARKET_CLOSED | The specified market is closed |
| UNSUPPORTED_MARKET_TYPE | Profit/Loss calculations are not supported for this market type. |

Get Market Traded Volume (exchange)

The API GetMarketTradedVolume service allows you to obtain all the current odds and matched amounts on a single runner in a particular event.

Service Details

SOAP Action: `getMarketTradedVolume`

Known Issues

If the optional currencyCode is not specified, getCompleteMarketPricesCompressed and getMarketTradeVolumeCompressed occasionally return prices using an incorrect currency. The workaround is to always pass the desired currency in the currencyCode field.

Input

The following table describes the parameters used for calling the GetMarketTradedVolume service.

Table 115: 1 instance of GetMarketTradedVolumeReq

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session |

| Parameter | Mandatory? | Type | Description |
|--------------|------------|---------|--|
| asianLineId | N | Integer | token (see Betfair Simple Data Types). Mandatory if the market specified by Market ID is an Asian Market, otherwise optional |
| currencyCode | N | String | Three letter ISO 4217 code |
| marketId | Y | Integer | The desired market |
| selectionId | Y | Integer | The desired runner id |

Output

The following table describes the parameters returned from the GetMarketTradedVolume service.

Table 116: 1 Instance of GetMarketTradedVolumeResp

| Parameter | Type | Description |
|----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| errorCode | see ... | Specific error code (see enum below) |
| minorErrorCode | String | Reserved for future use - currently always null |
| priceItems | ArrayOfVolumeInfo | Odds and matched totals data. |
| actualBSP | Double | Post-reconciliation, this is the Betfair Starting Price. Prior to reconciliation, this field contains zero. |

The GetMarketTradedVolume service returns priceItems as an array of VolumeInfo. The table below shows the fields in VolumeInfo

Table 117: ArrayOfVolumeInfo 0 or more instances of VolumeInfo

| Parameter | Type | Description |
|----------------------------|--------|---|
| odds | Double | Odds on the selection |
| totalMatchedAmount | Double | Total amount matched for the given odds |
| totalBspBackMatchedAmmount | Double | Post-reconciliation, this is the total amount of BSP back bets matched. Pre-reconciliation, and for any odds not at the starting price, this field contains zero. |

| Parameter | Type | Description |
|--------------------------------|--------|--|
| totalBspLiabilityMatchedAmount | Double | Post-reconciliation, this is the total amount of liability matched for BSP lay bets. Pre-reconciliation, and for any odds not at the starting price, this field contains zero. |

**Note:**

Type Double may contain numbers, INF, -INF, and NaN.

Error Codes

The following table describes the error codes returned by GetMarketTradedVolume.

Table 118: GetMarketTradedVolumeErrorEnum

| Value | Condition |
|---------------------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_ASIAN_LINE | The asian line specified does not exist |
| INVALID_CURRENCY | The currency code is not valid |
| INVALID_MARKET | The market Id is invalid. Check that you have sent your service request to the correct exchange server (the Australian exchange server cannot see markets on the UK exchange server, and vice versa). |
| INVALID_RUNNER | The runner id specified does not exist |
| MARKET_CLOSED | Market closed |
| MARKET_TYPE_NOT_SUPPORTED | The specified market ID corresponds to a market that is not supported for this service |
| NO_RESULTS | No results were returned for the request arguments |

Get Market Traded Volume Compressed (exchange)

The API GetMarketTradedVolumeCompressed service allows you to obtain the current price (odds) and matched amounts at each price on all of the runners in a particular market.

Service Details

SOAP Action: `getMarketTradedVolumeCompressed`

Input

The following table describes the parameters used for calling the GetMarketTradedVolumeCompressed service.

Table 119: 1 instance of `GetMarketTradedVolumeCompressedReq`

| Parameter | Mandatory? | Type | Description |
|--------------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| currencyCode | N | String | Three letter ISO 4217 code. |
| marketId | Y | Integer | The desired market |

Output

The following table describes the parameters returned from the `GetMarketTradedVolumeCompressed` service.

Table 120: 1 Instance of `GetMarketTradedVolumeCompressedResp`

| Parameter | Type | Description |
|----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| errorCode | see ... | Specific error code (see enum below) |
| minorErrorCode | String | Reserved for future use - currently always null |
| currencyCode | String | Three letter ISO 4217 code |
| marketId | Integer | The Id of the requested market |
| tradedVolume | String | The compressed string containing the traded volume |

The `GetMarketTradedVolumeCompressed` service returns `tradedVolume` as an array of delimited string. The table below shows the fields in

Table 121: `tradedVolume`

| Field No. | Type | Name | Notes |
|---|---------|--------------|---------------------------------------|
| Runner information fields. Each field is delimited by a ~ (tilda). Each runner is delimited by a : (colon) (0 or more iterations) | | | |
| 1 | Integer | selection Id | Id of the selection (runner). This Id |

| Field No. | Type | Name | Notes |
|--|---------|------------------------------------|--|
| 2 | Integer | Asian Line Id | is the same for the same selection across markets. The Asian Line Id, which is used in conjunction with the selection Id in Handicap markets. |
| 3 | Double | Actual BSP | In BDP markets, post-reconciliation, this is the Betfair Starting Price. Prior to reconciliation in BSP markets, this field contains zero. This will contain zero for all non BSP markets at all times. |
| 4 | Double | Total BSP Back Matched Amount | In BSP markets, post-reconciliation, this is the total amount of BSP back bets matched. Pre-reconciliation, this field contains zero. The field will contain zero for all non BSP markets at all times. |
| 5 | Double | Total BSP Liability Matched Amount | In BSP markets, post-reconciliation, this is the total amount of liability matched for BSP lay bets. Pre-reconciliation, this field contains zero. The field will contain zero for all non BSP markets at all times. |
| <p>Traded amounts. Each field is delimited by a ~ (tilda). Each traded amount is delimited by a (pipe) (0 or more iteration)</p> | | | |

| Field No. | Type | Name | Notes |
|-----------|--------|----------------------|---|
| 1 | Double | Odds | The odds for this traded amount |
| 2 | Double | Total Matched Amount | The total amount matched for the given odds |

**Note:**

Type Double may contain numbers, INF, -INF, and NaN.

Error Codes

The following table describes the error codes returned by GetMarketTradedVolumeCompressed.

Table 122: GetMarketTradedVolumeCompressedErrorEnum

| Value | Condition |
|---------------------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_CURRENCY | The currency code is not valid |
| INVALID_MARKET | The market Id is invalid. Check that you have sent your service request to the correct exchange server (the Australian exchange server cannot see markets on the UK exchange server, and vice versa). |
| EVENT_CLOSED | Market closed |
| EVENT_INACTIVE | The market is currently inactive |
| EVENT_SUSPENDED | The market is currently suspended |
| MARKET_TYPE_NOT_SUPPORTED | The MarketID supplied refers to a market that is not supported by the API. |

Get Private Markets (exchange)

The API GetPrivateMarkets service allows you to retrieve active and suspended private markets that are within an EventType that is not visible on Betfair.com or with the GetEvents or GetActiveEvents services.

Service Details

SOAP Action: `getPrivateMarkets`

Known Issues

All BF Tennis, Golf, Cricket and Rugby Union markets are showing a prefix of 'Group A/B/C/D' in the menuPath, while there is no such valid component.

Input

The following table describes the parameters used for calling the GetPrivateMarkets service.

Table 123: 1 instance of `GetPrivateMarketsReq`

| Parameter | Mandatory? | Type | Description |
|-------------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token. |
| EventTypeID | Y | Integer | The Id of the event holding the private markets. |
| marketType | Y | MarketTypeEnum | The type of markets you want returned. |
| locale | N | String | Specify the language for the reply if you want a different language than the account default. For more information, |

Output

The following table describes the parameters returned from the `GetPrivateMarkets` service.

Table 124: 1 Instance of `GetPrivateMarketsResp`

| Parameter | Type | Description |
|----------------|--|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| errorCode | GetPrivateMarketsErrorEnum | getMarket specific error code (see enum below) |
| privateMarkets | ArrayOfPrivateMarket | An array of private market data |
| minorErrorCode | String | Reserved for future use - currently always nulls |

The `GetPrivateMarkets` service returns an array named `PrivateMarket`. The table below shows the fields in `PrivateMarket`.

Table 125: `ArrayOfPrivateMarket`

| Parameter | Type | Description |
|----------------|-----------------------------|---|
| name | String | The localised name of the market |
| marketId | Integer | The Id of the private market. |
| menuPath | String | The detailed path through the Betfair menu to reach this market |
| eventHierarchy | <code>ArrayOfEventId</code> | The eventHierarchy is an array containing the EventIds that |

| Parameter | Type | Description |
|-----------|------|--|
| | | lead from the top of the menu path to the <code>marketId</code> of the private market. |

Error Code

The following table describes the error codes returned by `GetPrivateMarkets`.

Table 126: GetPrivateMarketsErrorEnum

| Value | Condition |
|-----------------------|--|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader <code>errorCode</code> field for additional information |
| INVALID_LOCALE | The locale string was not recognized. Returned results are in English |
| INVALID_EVENT_TYPE_ID | The event type supplied is not a hidden event |
| INVALID_MARKET_TYPE | The market type does not exist |
| NO_RESULTS | There are no active or suspended private markets in that event type |

Get Silks (exchange)

The API `GetSilks` service allows you to retrieve a relative URL to the jockey silk image and data about each selection including:

- Silks description
- Trainer name
- Age and Weight
- Form
- Days since last run
- Jockey claim
- Wearing text
- Saddle cloth
- Stall draw



Note:

The URL returned in the response is relative to a base URL. The base URL for the silks image is:

http://content-cache.betfair.com/feeds_images/Horses/SilkColours/

Service Details

SOAP Action: `getSilks`

Input

The following table describes the parameters used for calling the `GetSilks` service.

Table 127: 1 instance of `GetSilksReq`

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|--|
| header | Y | APIRequestHeader | The API Request Header that contains the session token (see API Request/Response Header). |
| locale | N | String | The locale to use when returning results. If not specified, the default locale for the user's account is used. |
| markets | Y | ArrayOfInt | An array of market Ids (up to 40) |

Output

The following table describes the parameters returned from the GetSilks service.

Table 128: 1 Instance of `GetSilksResp`

| Parameter | Type | Description |
|----------------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header (see API Request/Response Header). |
| errorCode | see ... | GetSilks specific error code (see enum below) |
| marketDisplayDetails | see ... | The market data |
| minorErrorCode | String | Reserved for future use - currently always nulls |

The GetSilks service returns an array of type `MarketDisplayDetail`. The table below shows the fields in `MarketDisplayDetail`

Table 129: `ArrayOfMarketDisplayDetail`

| Parameter | Type | Description |
|-------------|-------------------------|---|
| errorCode | see ... | The errorCode specific to this market |
| marketId | Integer | The marketId containing the display details below |
| racingSilks | see ... | An array containing the details for each runner |

The table below shows the fields in `RacingSilk`

Table 130: ArrayOfRacingSilk

| Parameter | Type | Description |
|-------------|---------|--|
| selectionId | Integer | The selection Id of the runner |
| silksURL | String | A relative URL to an image file corresponding to the jockey silk. This is the same image displayed on the Betfair.com website. You must add the value of this field to the base URL: http://content-cache.betfair.com/feeds_images/Horses/SilkColours/ |
| silksText | String | The textual description of the jockey silk |
| trainerName | String | The name of the horse's trainer |
| ageWeight | String | The age and weight of the horse |
| form | String | The form identifier |
| daysSince | Integer | The number of days since the horse's last run |
| jockeyClaim | Integer | The jockey's claim |
| wearing | String | Any extra equipment the horse is wearing |
| saddleCloth | Integer | The number on the saddle |
| stallDraw | Integer | The stall number the horse is starting from |

Error Code

The following table describes the error codes returned by GetSilks.

Table 131: GetSilksErrorEnum

| Value | Condition |
|---------------------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_LOCALE | The locale string was not recognized. |
| INVALID_NUMBER_OF_MARKETS | You have specified no markets, or more than 40 markets. |

Table 132: MarketDisplayErrorEnum

| Value | Condition |
|-------|-----------|
| OK | |

| Value | Condition |
|--------------------|---|
| INVALID_MARKET | Invalid Market ID supplied. Make sure you have sent the request to the correct exchange server. Check that you have sent your service request to the correct exchange server (the Australian exchange server cannot see markets on the UK exchange server, and vice versa). |
| NO_SILKS_AVAILABLE | The exchange does not publish silks for this market. |

Get Silks V2 (exchange)

The API GetSilksV2 service allows you to retrieve a relative URL to the jockey silk image and data about each selection including:

- Age and Weight
- bred
- Colour type
- Dam Bred
- Dam Name
- DamSire Bred
- DamSire Name
- DamSire Year Born
- Days since last run
- ForecastPrice Price Denominator
- ForecastPrice Price Numerator
- Form
- Jockey claim
- Jockey Name
- Official Rating
- Owner name
- Saddle cloth
- Sex type
- Silks description
- Sire Bred
- Sire Name
- Sire Year Born
- Stall draw
- Trainer name
- Wearing text

This service returns all of the data as the existing getSilks service, but adds additional information.



Note:

The URL returned in the response is relative to a base URL. The base URL for the silks image is:

http://content-cache.betfair.com/feeds_images/Horses/SilkColours/

Service Details

SOAP Action: getSilksV2

Input

The following table describes the parameters used for calling the GetSilksV2 service.

Table 133: 1 instance of GetSilksV2Req

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|--|
| header | Y | APIRequestHeader | The API Request Header that contains the session token (see API Request/Response Header). |
| locale | N | String | The locale to use when returning results. If not specified, the default locale for the user's account is used. |
| markets | Y | ArrayOfInt | An array of market Ids (up to 40) |

Output

The following table describes the parameters returned from the GetSilksV2 service.

Table 134: 1 Instance of GetSilksV2Resp

| Parameter | Type | Description |
|----------------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header (see API Request/Response Header). |
| errorCode | see ... | GetSilks specific error code (see enum below) |
| marketDisplayDetails | see ... | The market data |
| minorErrorCode | String | Reserved for future use - currently always nulls |

The GetSilks service returns an array of type MarketDisplayDetailV2. The table below shows the fields in MarketDisplayDetailV2

Table 135: ArrayOfMarketDisplayDetailV2

| Parameter | Type | Description |
|-------------|-------------------------|---|
| errorCode | see ... | The errorCode specific to this market |
| marketId | Integer | The marketId containing the display details below |
| racingsilks | see ... | An array containing the details for each runner |

The table below shows the fields in RacingSilkV2

Table 136: ArrayOfRacingSilkV2

| Parameter | Type | Description |
|--------------------------|-------------------------|--|
| selectionId | Integer | The selection Id of the runner |
| silksURL | String | A relative URL to an image file corresponding to the jockey silk. This is the same image displayed on the Betfair.com website. You must add the value of this field to the base URL: http://content-cache.betfair.com/feeds_images/Horses/SilkColours/ |
| silksText | String | The textual description of the jokey silk |
| trainerName | String | The name of the horse's trainer |
| ageWeight | String | The age and weight of the horse |
| form | String | The form identifier |
| daysSince | Integer | The number of days since the horse's last run |
| jockeyClaim | Integer | The jockey's claim |
| wearing | String | Any extra equipment the horse is wearing |
| saddleCloth | Integer | The number on the saddle |
| stallDraw | Integer | The stall number the horse is starting from |
| ownerName | String | The name of the horse's owner |
| jockeyName | String | The name of the jockey |
| colour | String | The colour of the horse |
| sex | String | The sex of the horse |
| forecastPriceNumerator | Integer | The forecast price numerator |
| forecastPriceDenominator | Integer | The forecast price denominator |
| officialRating | Integer | The official rating |
| sire | see ... | Information on the horse's father |
| dam | see ... | Information on the horse's mother |
| damSire | see ... | Information on the horse's grandfather on the mother's side |

The following tables describes the fields in Breeding

Table 137: Breeding

| Parameter | Type | Description |
|-----------|--------|-----------------------------|
| name | String | The horse's name |
| bred | String | The horse's bred |
| yearBorn | String | The year the horse was born |

Error Code

The following table describes the error codes returned by GetSilks.

Table 138: GetSilksErrorEnum

| Value | Condition |
|---------------------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_LOCALE | The locale string was not recognized. |
| INVALID_NUMBER_OF_MARKETS | You have specified no markets, or more than 40 markets. |

Table 139: MarketDisplayErrorEnum

| Value | Condition |
|--------------------|---|
| OK | |
| INVALID_MARKET | Invalid Market ID supplied. Make sure you have sent the request to the correct exchange server. Check that you have sent your service request to the correct exchange server (the Australian exchange server cannot see markets on the UK exchange server, and vice versa). |
| NO_SILKS_AVAILABLE | The exchange does not publish silks for this market. |

Bet Placement API Services Reference

Cancel Bets (exchange)

The API CancelBets service allows you to cancel multiple unmatched (1 to 40) bets placed on a single Market. There is an instance of `CancelBetsResp` returned in the output for each instance of `CancelBets` in the input. The success or failure of the individual bet cancellation operation will be indicated by the `Success` Boolean. If a portion of the original bet is already matched, `cancelBets` cancels the unmatched portion of the bet.

Note:



The funds from cancelled bets will be restored to the local wallet for the exchange server that you placed the bet on. If you placed the bet on the UK exchange server, your UK wallet will be credited. If you placed it on the Australian exchange server, your Australian wallet will be credited.

This service is permitted even if the local wallet is not active (in other words, even if it is suspended or in some other non-active state).

Service Details

SOAP Action: `cancelBets`

Input

The following table describes the parameters used for calling the `CancelBets` service.

Table 140: 1 instance of `CancelBetsReq`

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| bets | Y | <code>ArrayOfCancelBets</code> | Array of between 1 and 40 <code>CancelBets</code> |

The `CancelBets` service requires `bets` as an array of `CancelBets`. The table below shows the fields in `CancelBets`

Table 141: `ArrayOfCancelBets` 1 to 40 instances of `CancelBets`

| Parameter | Type | Description |
|-----------|------|--|
| betId | Long | Unique identifier of bet to be cancelled |

Output

The following table describes the parameters returned from the `CancelBets` service.

Table 142: 1 Instance of `CancelBetsResp`

| Parameter | Type | Description |
|----------------|-----------------------------------|--|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| betResults | ArrayOfCancelBetsResult | Between 1 and 40 CancelBetsResults items |
| errorCode | see ... | Specific error code (see below) |
| minorErrorCode | String | If <code>API_ERROR</code> or <code>BET_NOT_CANCELLED</code> is returned, this field contains <code>age.verification.required</code> if the user has not completed the age verification process with Betfair. |

The CancelBets service returns `betResults` as an array of `CancelBetsResult`. The table below shows the fields in `CancelBetsResult`

Table 143: ArrayOfCancelBetsResult 1 to 40 instances of `CancelBetsResult`

| Parameter | Type | Description |
|---------------|-------------------------|--|
| betId | Long | The unique bet identifier |
| resultCode | see ... | Further information about the success or failure of the bet cancellation |
| sizeCancelled | Double | Amount cancelled |
| sizeMatched | Double | Amount of original bet matched since placement |
| success | Boolean | If true the bet was successfully cancelled |

Error Codes

The following table describes the error codes returned by CancelBets.

Table 144: `CancelBetsErrorEnum`

| Value | Condition |
|------------------------|---|
| OK | |
| <code>API_ERROR</code> | General API Error. One reason you might receive this message is that you have attempted to place a bet on the Australian exchange server but your Australian wallet is suspended because Betfair have not yet received confirmation of your name and address. Check the APIResponseHeader <code>errorCode</code> field for additional information |

| Value | Condition |
|--------------------------------|--|
| BET_IN_PROGRESS | The exchange has not completed cancelling the bet before the bet processing time expired. You should use GetMUBets to determine if the bet was cancelled successfully. |
| BBM_DAEMON_NOT_AVAILABLE | The exchange encountered a critical error and was not able to cancel your bet. Your bet was not cancelled. |
| INVALID_MARKET_ID | The bets were not all from the same market. |
| INVALID_NUMER_OF_CANCELLATIONS | Number of bets < 1 or > 40 |
| MARKET_IDS_DONT_MATCH | Bet ID does not exist |
| MARKET_STATUS_INVALID | The status of the Market is invalid for this action. The Market may be suspended or closed. |
| MARKET_TYPE_NOT_SUPPORTED | Invalid Market type |

The following table lists the possible error codes for a particular bet.

Table 145: CancelBetsResultEnum

| Value | Condition |
|-------------------------------|---|
| OK | |
| BBM_DAEMON_NOT_AVAILABLE | The exchange encountered a critical error and was not able to cancel your bet. Your bet was not cancelled. |
| BET_IN_PROGRESS | The exchange has not completed cancelling the bet before the bet processing time expired. You should use GetMUBets to determine if the bet was cancelled successfully. |
| BET_NOT_CANCELLED | Bet not cancelled. The exchange has not cancelled your bet(s). This may be because the account is suspended. |
| BSD_BETS_CANNOT_BE_CANCELLED | The betId corresponds to a BSP bet. BSP bets cannot be cancelled once placed. |
| FROM_COUNTRY_FORBIDDEN | Cancellation request from restricted country |
| INPLAY_FORBIDDEN | Not used. |
| INPLAY_FROM_COUNTRY_FORBIDDEN | Cancellation request for in-play event |
| INVALID_BET_ID | Not Used |
| REMAINING_CANCELLED | The bet has been cancelled. Check sizeMatched and sizeCancelled to determine if any portion of the bet was matched prior to cancelling. |
| SITE_UPGRADE | The site is not available while it is being upgraded |
| TAKEN_OR_LAPSED | Bet already matched, lapsed, or the Bet ID does not exist. One reason you might receive this message is that you have sent the request to the wrong exchange server. The UK exchange server |

| Value | Condition |
|---------------|--|
| | cannot return information about bets placed on the Australian exchange server, and vice versa. |
| UNKNOWN_ERROR | Internal error. Bet not cancelled. |

**Note:**

If success is false then all bet cancellations have failed.

If success is true then size of betResults equals the size of bets.

Cancel Bets By Market (exchange)

The API CancelBetsByMarket service allows you to cancel all unmatched bets (or unmatched portions of bets) placed on one or more Markets. You might use this service to quickly close out a position on a market.

**Note:**

You do not receive any indication if a portion of one or more bets on a market have been partially matched. If you want to cancel bets and have the API inform you if a bet was partially matched, use the CancelBets service instead.

**Note:**

The funds from cancelled bets will be restored to the local wallet for the exchange server that you placed the bet on. If you placed the bet on the UK exchange server, your UK wallet will be credited. If you placed it on the Australian exchange server, your Australian wallet will be credited.

This service is permitted even if the local wallet is not active (in other words, even if it is suspended or in some other non-active state).

Service Details

SOAP Action: `cancelBetsByMarket`

Known Issues

Does not cancel any bets in a market in which the user has an SP bet. Use the cancelBets service instead.

Input

The following table describes the parameters used for calling the CancelBetsByMarket service.

Table 146: 1 instance of `CancelBetsByMarketReq`

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| markets | Y | ArrayOfInts | An array of integers representing marketIds |

Output

The following table describes the parameters returned from the CancelBetsByMarket service.

Table 147: 1 Instance of CancelBetsByMarketResp

| Parameter | Type | Description |
|----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| results | ArrayOfCancelBetsByMarketResult | Array of CancelBetsByMarketResults, one for each submitted marketId |
| errorCode | see ... | Specific error code (see below) |
| minorErrorCode | String | If API_ERROR or BET_NOT_CANCELLED is returned, this field contains age.verification.required if the user has not completed the age verification process with Betfair. |

The CancelBetsByMarket service returns results as an array of CancelBetsByMarketResult. The table below shows the fields in CancelBetsByMarketResult

Table 148: ArrayOfCancelBetsByMarketResult 1 to n instances of CancelBetsByMarketResult

| Parameter | Type | Description |
|------------|-------------------------|--|
| marketId | Integer | The market Id that this response corresponds to |
| resultCode | see ... | Further information about the success or failure of the bet cancellation |

Error Codes

The following table describes the error codes returned by CancelBetsByMarket.

Table 149: CancelBetByMarketErrorEnum

| Value | Condition |
|-----------|---|
| OK | There were no general errors. Check the results for each CancelBetsByMarketResult |
| API_ERROR | General API Error. One reason you might receive this message is that you have attempted to cancel a bet on the Australian exchange server but your Australian wallet is suspended because Betfair have not yet received confirmation of your name |

| Value | Condition |
|---------------------------|---|
| INVALID_NUMBER_OF_MARKETS | and address. Check the APIResponseHeader errorCode field for additional information You supplied a null value or zero number of market IDs |

The following table lists the possible error codes for a particular market.

Table 150: CancelBetsByMarketResultEnum

| Value | Condition |
|-------------------------------|--|
| OK | |
| BET_NOT_CANCELLED | Unspecified error. Bet not cancelled. |
| FROM_COUNTRY_FORBIDDEN | Cancellation request from restricted country |
| INPLAY_FORBIDDEN | Not used. |
| INPLAY_FROM_COUNTRY_FORBIDDEN | Cancellation request for in-play event from a country where in-play betting is not allowed |
| INVALID_MARKET | marketId does not exist. One reason you might receive this message is that you have sent the request to the wrong exchange server. The UK exchange server cannot return information about bets placed on the Australian exchange server, and vice versa. |
| SITE_UPGRADE | Betting is not available while the web site is being upgraded |
| UNKNOWN_ERROR | Internal error. Bets not cancelled. |
| MARKET_STATUS_INVALID | The status of the Market is invalid for this action. The Market may be suspended or closed. |
| NO_UNMATCHED_BETS | The market does not contain any unmatched bets. The bets may have been matched or lapsed. |

Place Bets (exchange)

The API PlaceBets service allows you to place multiple (1 to 60) bets on a single Market. There is an instance of PlaceBetsResp returned in the output for each instance of PlaceBets in the input. The success or failure of the individual bet placement operation is indicated by the Success Boolean.



Note:

To bet on an event, you need to have sufficient funds available in the relevant local wallet to cover your entire liability. If you want to bet on the Australian exchange server and you do not have sufficient funds in your Australian wallet to cover the liability, you must first transfer funds into that wallet from your UK wallet by using the transferFunds service (see [Betfair Simple Data Types](#)). Also, your wallet must be active (and not, for example, suspended for any reason) at the time you place the bet.

Bet Types

You can specify, for each bet, if you want to place a Back bet or a Lay bet.

- Back bets win when the selection is settled as the winner in the market.

- Lay bets win when the selection is settled as a loser in the market.

For more information on Bet types, see the Betfair website help.

Bet Categories

You can specify, for each bet, whether the bet is a regular exchange bet, or a Betfair Market on Close (or Starting Price) bet (with or without a price limit).

- Exchange bets are placed on the market and are matched against bets at the specified or better price. Exchange bets are matched on a first in, first matched basis.
- Market on Close (MOC) 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.
- Limit on Close (LOC) 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)

The required fields in `bets` are dependent on the category of bet. The following table shows the required fields for each bet category.

Table 151: Valid Bet Category request field combinations

| Bet Category | Price | Size | BspLiability | BetPersistenceType |
|-----------------|-------|------|--------------|--------------------|
| Exchange | Yes | Yes | No | Yes |
| Market on Close | No | No | Yes | No |
| Limit on Close | Yes | No | Yes | No |

Bet Persistence

You can specify what happens to an Exchange bet that is unmatched when the market is reconciled and the starting price is calculated.

- None - The unmatched bet is cancelled when the market is reconciled and turned in-play.
- IP - The unmatched bet stays as an unmatched bet when the market is turn in-play.
- SP - The unmatched bet becomes a Market on Close bet and is matched at the starting price.

Service Details

SOAP Action: `placeBets`

Known Issues

When calling `placeBets` with multiple bets and a mix of valid & invalid size values, success is 'true' and `resultCode` is 'OK' for bets with valid size. However, no `betIds` are returned and no bets are actually placed. The bet(s) with an invalid size requested correctly returns 'INVALID_SIZE'. When calling `placeBets` and one of the bets result in an `INVALID_INCREMENT` error, this `resultCode` is also given to all of the 'valid' bets as well.

When calling `placeBets`, and one of the bets result in an `INVALID_INCREMENT` error, this `errorCode` is also given to all of the 'valid' bet results as well.

When placing an Exchange bet, if BspLiability field value is null, then the bet is not placed, but no error is returned.
When placing an exchange bet, use 0.0 as the value of the BspLiability field.

Input

The following table describes the parameters used for calling the PlaceBets service.

Table 152: 1 instance of PlaceBetsReq

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| bets | Y | ArrayOfPlaceBets | Array of Bet containing details of the bets to place |

The PlaceBets service requires `bets` as an array of PlaceBets. The table below shows the fields in PlaceBets

Table 153: ArrayOfPlaceBets 1 to 60 instances of PlaceBets

| Parameter | Type | Description |
|--------------------|--|--|
| asianLineId | Integer | The ID of the Asian Handicap market to place bets on. Set to 0 for non-Asian handicap Markets. |
| betCategoryType | BetCategoryTypeEnum | Specify if the bet is an Exchange bet, Market on Close SP bet or Limit on Close SP bet. If you specify Limit on Close, specify the desired limit using the <code>price</code> field. |
| betPersistenceType | BetPersistenceTypeEnum | Specify what happens to an unmatched (or partially unmatched) exchange bet when the market turns in-play. If <code>betCategoryType</code> is an SP bet, <code>betPersistenceType</code> must be set to NONE. |
| betType | BetTypeEnum | Bet type (back, lay or equivalent Asian Handicap types) |
| bspLiability | Double | This is the maximum amount of money you want to risk for a BSP bet. The minimum amount for a back bet is £2 (or equivalent). The minimum amount for a lay bet is £10 (or equivalent) For a back bet, this is equivalent to the |

| Parameter | Type | Description |
|-------------|---------|---|
| | | stake on a normal exchange bet. For a lay bet, this is the equivalent to the liability on a normal exchange bet. If after the market is reconciled, the actual stake is calculated once the price is known. |
| marketId | Integer | The ID of the market |
| price | Double | The price (odds) you want to set for the bet. Valid values are 1.01 to 1000. For a BSP Limit on Close bet, specify the desired price limit. For a Back bet, the minimum price you want. If the Starting Price is lower than this amount, then your bet is not matched. For a Lay bet, the maximum acceptable price. If the Starting Price is higher than this amount, then your bet is not matched. 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). |
| selectionId | Integer | ID of the desired runner or selection within the market |
| size | Double | The stake (amount) for an exchange bet. The minimum amount for a back bet is £2 (or equivalent). If the betPersistenceType is set to SP, then the minimum amount for a lay bet is £10 (or equivalent), otherwise, the minimum lay bet amount is £2 (or equivalent). |

Output

The following table describes the parameters returned from the PlaceBets service.

Table 154: 1 Instance of PlaceBetsResp

| Parameter | Type | Description |
|-----------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session |

| Parameter | Type | Description |
|----------------|------------------------------------|---|
| betResults | ArrayOfPlaceBetsResult | token (see “API Request/Response Header” on page 20). Information about individual bet placement results (or errors). If any bet request results in an error, then none of the bets are placed. |
| errorCode | PlaceBetsErrorEnum | An overall error message, usually related to the format of the request. Errors for an individual bet are contained in the individual BetPlacementResult |
| minorErrorCode | String | If ACCOUNT_SUSPENDED is returned, this field contains age.verification.required if the user has not completed the age verification process with Betfair. |

The PlaceBets service returns `betResults` as an array of `PlaceBetsResult`. The table below shows the fields in `PlaceBetsResult`.

Table 155: ArrayOfBetPlacementResult 1 to 60 instances of PlaceBetsResult

| Parameter | Type | Description |
|---------------------|-------------------------------------|--|
| averagePriceMatched | Double | Average price taken |
| betId | Long | The unique identifier for the bet |
| resultCode | PlaceBetsResultEnum | Further information about the success or failure of the bet placement. |
| sizeMatched | Double | For an exchange bet, the actual price taken |
| success | Boolean | True if bet successfully placed, otherwise false. |

Error Codes

The following table describes the error codes returned by PlaceBets.

Table 156: PlaceBetsErrorEnum

| Value | Condition |
|-------|---|
| OK | Note: You should also check the individual PlaceBetsResultEnum for each instance of PlaceBetsResult |

| Value | Condition |
|--------------------------------|--|
| ACCOUNT_CLOSED | The user's account has been closed. Please contact Betfair |
| ACCOUNT_SUSPENDED | Account has been suspended. One reason you might receive this message is that you have attempted to place a bet on the Australian exchange server but your Australian wallet is suspended because Betfair have not yet received confirmation of your name and address. |
| API_ERROR | General API error. Check the APIResponseHeader errorCode for additional information |
| AUTHORISATION_PENDING | Account is pending authorisation. If the PlaceBetsResultEnum is also CANNOT_ACCEPT_BET, this means that the market is operating under the Tasmanian Gaming Commission rules and the account holder's identity has not yet been verified. |
| BACK_LAY_COMBINATION | Bets contains a Back and a Lay on the same runner and the Back price is less then or equal to the Lay price. |
| BBM_DAEMON_NOT_AVAILABLE | The exchange encountered a critical error and was not able to match your bet. Your bet was not placed. |
| BET_IN_PROGRESS | The exchange has not completed matching the bet before the bet matching time expired. The BET_IN_PROGRESS error is rarely returned but is an indication that there is an internal issue in placing the bet. The placement of the bet is likely but not guaranteed. If the BET_IN_PROGRESS error is returned we recommend that you proceed with caution. Following a BET_IN_PROGRESS error, and the bet is not returned following a call to getMUBets for more than 1 second: If prices appear to be moving, it indicates that the system is working as placing/matching is occurring. If following a second request returns BET_IN_PROGRESS then stop betting and check the system status. |
| BETWEEN_1_AND_60_BETS_REQUIRED | Number of BetPlacement less than 1 or greater than 60 |
| BSP_BETTING_NOT_SUPPORTED | You have requested a BSP bet, but BSP betting is not supported for this market. |
| DIFFERING_MARKETS | All bets not all for the same market |
| EVENT_CLOSED | Market has already closed |
| EVENT_INACTIVE | Market is not active |
| EVENT_SUSPENDED | Market is suspended |
| FROM_COUNTRY_FORBIDDEN | Bet origin from a restricted country |
| INTERNAL_ERROR | Internal error occurred |

| Value | Condition |
|---------------------------|---|
| INVALID_MARKET | MarketID doesn't exist. Check that you have sent your service request to the correct exchange server (the Australian exchange server cannot see markets on the UK exchange server, and vice versa). |
| MARKET_TYPE_NOT_SUPPORTED | Market Type is invalid or does not exist |
| SITE_UPGRADE | Site is currently being upgraded |

Table 157: PlaceBetsResultEnum

| Value | Condition |
|--------------------------|---|
| OK | |
| ACCOUNT_CLOSED | Account is closed - please contact the Betfair Helpdesk |
| ACCOUNT_IS_EXCLUDED | The account has been locked due to self-exclusion |
| ACCOUNT_LOCKED | Account is locked - please contact the Betfair Helpdesk |
| ACCOUNT_SUSPENDED | Account has been suspended. One reason you might receive this message is that you have attempted to place a bet on the Australian exchange server but your Australian wallet is suspended because Betfair have not yet received confirmation of your name and address. |
| AUTHORISATION_PENDING | Account is pending authorisation. If the PlaceBetsResultEnum is also CANNOT_ACCEPT_BET, this means that the market is operating under the Tasmanian Gaming Commission rules and the account holder's identity has not yet been verified. |
| BBM_DAEMON_NOT_AVAILABLE | The exchange encountered a critical error and was not able to match your bet. Your bet was not placed. |
| BET_IN_PROGRESS | The exchange has not completed processing the bet before the time expired. The BET_IN_PROGRESS error is rarely returned but is an indication that there is an internal issue in placing the bet. The placement of the bet is likely but not guaranteed. If the BET_IN_PROGRESS error is returned we recommend that you proceed with caution. Following a BET_IN_PROGRESS error, and the bet is not returned following a call to getMUBets for more than 1 second: If prices appear to be moving, it indicates that the system is working as placing/matching is occurring. If following a second request returns BET_IN_PROGRESS then stop betting and check the system status. |

| Value | Condition |
|--|---|
| BSP_BETTING_NOT_ALLOWED | You are not allowed to place a BSP bet. |
| BSP_BETTING_NOT_SUPPORTED | You have requested a BSP bet, but BSP betting is not supported for this market. |
| BSP_MULTIPLE_BETS_NOT_ALLOWED | Bets contains a Back and a Lay on the same runner |
| CANNOT_ACCEPT_BET | Your Bet cannot be accepted. This may be due to the bet size or, if the PlaceBetsErrorEnum is AUTHORISATION_PENDING, the market is under the Australian Gaming Commission rules and the account holder's identity has not been verified. |
| EVENT_CLOSED | Market has already closed |
| EVENT_RECONCILED | The market has already been reconciled and you attempted to place an SP bet. |
| EXPOSURE_CALCULATION_IN_PROGRESS | The bet was not placed because the system is calculating your exposure. This error is most likely to occur when the account is close to the exposure limit or the available to bet balance is near zero, and you are placing/updating multiple bets in a short space of time. |
| EXPOSURE_OR_AVAILABLE_BALANCE_EXCEEDED | Account Exceeded Exposure Limit or Available to Bet Balance |
| FROM_COUNTRY_ON_EVENT_FORBIDDEN | Bet origin from a country on a restricted event |
| INPLAY_FROM_COUNTRY_FORBIDDEN | Bet originates from a restricted country on an in play event |
| INSUFFICIENT_BALANCE | Insufficient available money in the account |
| INVALID_ASIAN_LINE_ID | Asian Handicap ID doesn't exist |
| INVALID_BET_TYPE | Bet Type is invalid or does not exist |
| INVALID_INCREMENT | The Price is not in allowable increments |
| INVALID_MARKET | MarketID does not exist |
| INVALID_PERSISTENCE | The betPersistenceType field contains a value that is not recognised, you have specified a value other than NONE for an SP bet or on a market that is not turning in-play. |
| INVALID_PRICE | Price is out of allowable range |
| INVALID_SELECTION | The selection is invalid or does not exist |
| INVALID_SIZE | The stake is invalid or does not exist. Make sure the stake size is equal to or greater than the minimum allowed stake size of £2 (£0.01 for Range markets). |
| LINES_OUT_OF_RANGE | The lines are in the wrong range and increments |
| LOSS_LIMIT_EXCEEDED | The potential loss for the bet exceeds the loss limit for the account |
| SELECTION_REMOVED | The runner has been removed from the event |

| Value | Condition |
|---------------|---|
| UNKNOWN_ERROR | An unknown error occurred for this bet. Check the <code>errorCode</code> and <code>minorErrorCode</code> fields in <code>PlaceBetErrorEnum</code> . |
| VACANT_TRAP | The selected trap is vacant |

Determining if your bet was placed

You can reliably determine if a bet was successfully placed by checking for the presence of the `betId` field. If no `betId` field is returned in the response, then the bet was not placed.

If you submit multiple bets in the `PlaceBets` request, then all bets must be successful or none of the bets are placed. If any single bet request is not successful, then all bet requests are discarded.

If the bet request fails due to a problem with an individual bet, you can view the reason for the failure in the `PlaceBetsResult resultCode` field.



Note:

Betting in-play is not allowed from Australia. If you place a bet that is set to persist in-play you cannot edit the bet once it turns in-play. You can cancel the bet after it has turned in-play (if it remains unmatched).



Note:

Betting on events other than horse racing from Holland is not allowed.

Update Bets (exchange)

The API `UpdateBets` service allows you to edit multiple (1 to 15) bets on a single Market. There is an instance of `UpdateBetsResp` returned in the output for each instance of `UpdateBets` in the input. The success or failure of the individual bet editing operation is indicated by the `Success Boolean`.



Note:

To update a bet, you need to have sufficient funds available in the relevant local wallet to cover your entire liability. If you want to update a bet on the Australian exchange server and you do not have sufficient funds in your Australian wallet to cover the new liability, you must first transfer funds into that wallet from your UK wallet by using the `transferFunds` service (see [Chapter 54](#)). Also, your wallet must be active (and not, for example, suspended for any reason) at the time you send the `UpdateBets` request.



Note:

If `newPrice` and `newSize` are both specified the `newSize` value is ignored. For example, an original bet is placed for £100 with odds of 1.5: `UpdateBets` is called with `newSize = 200`, `newPrice = 2`. The original bet is cancelled and a new bet is placed for £100 with odds of 2.

In-Play Delay Rules

The in-play delay applied for editing a bet depends on the specific change you make to the bet. The following table shows the possible ways to edit a bet and the amount of delay for each:

Table 158: Editing Bets and Resulting Time Delay

| If you... | The resulting delay is: |
|--------------------|--|
| Reduce the stake | None. The edit happens immediately. |
| Increase the stake | The delay is applied. |
| Change the price | The original bet is cancelled immediately. A new bet at the new price is placed after the delay. |

Service Details

SOAP Action: `updateBets`

Input

The following table describes the parameters used for calling the UpdateBets service.

Table 159: 1 instance of `UpdateBetsReq`

| Parameter | Mandatory? | Type | Description |
|---------------------|------------|----------------------------------|---|
| <code>header</code> | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| <code>bets</code> | Y | <code>ArrayOfUpdateBets</code> | Details of Bets to be edited |

The UpdateBets service requires `bets` as an array of UpdateBets. The table below shows the fields in UpdateBets

Table 160: ArrayOfUpdateBets 1 to 15 instances of UpdateBets

| Parameter | Type | Description |
|------------------------------------|--|---|
| <code>betId</code> | Long | The unique identifier for the bet |
| <code>newBetPersistenceType</code> | BetPersistenceTypeEnum | New persistence type on an exchange bet. Only valid before the market turns in-play. |
| <code>newPrice</code> | Double | New odds desired on the bet For BSP Limit on Close bets, newPrice should be set to the new limit desired. For BSP Limit on Close back bets, you can only change the limit downward. For BSP Limit on Close lay bets, you can only change the limit upward. |
| <code>newSize</code> | Double | New stake desired on the bet |
| <code>oldBetPersistenceType</code> | BetPersistenceTypeEnum | Original persistence type on an exchange bet. Only valid before the market turns in-play. |
| <code>oldPrice</code> | Double | For an exchange bet, original odds on the bet. |
| <code>oldSize</code> | Double | For an exchange bet, original stake on the bet |

Output

The following table describes the parameters returned from the UpdateBets service.

Table 161: 1 Instance of UpdateBetsResp

| Parameter | Type | Description |
|----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| betResults | ArrayOfUpdateBetsResult | Details of results. See below. |
| errorCode | see ... | Specific error code (see below) |
| minorErrorCode | String | If API_ERROR is returned, check for the following minorErrorCode responses: If ACCOUNT_SUSPENDED is returned, this field contains age.verification.required if the user has not completed the age verification process with Betfair. |

The UpdateBets service returns betResults as an array of UpdateBetsResult. The table below shows the fields in UpdateBetsResult

Table 162: ArrayOfUpdateBetsResult 1 to 15 instances of UpdateBetsResult

| Parameter | Type | Description |
|---------------|---------------------------------------|--|
| betId | Long | Original bet identifier |
| newPrice | Double | Price requested in all cases |
| newBetId | Integer | Id of any new bet created by update. Only if stake increased or odds changed otherwise set to 0 |
| newSize | Double | If new bet has been created, the size (stake) of the new bet. If no new bet created the new size of the original bet (doesn't include any amounts matched) |
| resultCode | UpdateBetsResultsEnum | Further information about the success or failure of the bet edit |
| sizeCancelled | Double | Any amount of the original bet cancelled as a result of the update request |
| success | Boolean | True if bet edit was successful. Otherwise false |

Error Codes

The following table describes the error codes returned by UpdateBets.

Table 163: UpdateBetsErrorEnum

| Value | Condition |
|---------------------------|---|
| OK | |
| ACCOUNT_CLOSED | The user's account is closed |
| ACCOUNT_PENDING | The user's account is pending authorisation. |
| ACCOUNT_SUSPENDED | Bet could not be cancelled. This may be because the user's local wallet is suspended (for example, because Betfair have not yet received confirmation of your name and address). Part of the process of updating a bet involves cancellation of the original bet. However, if between the original bet and the UpdateBets request the local wallet that was used to fund the bet has been suspended, then the original bet will not be cancelled and the bet update will therefore not be successful. |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| BBM_DAEMON_NOT_AVAILABLE | The exchange encountered a critical error and was not able to match your bet. Your bet was not placed. |
| BET_IN_PROGRESS | The exchange has not completed matching the bet before the bet matching time expired. You should use GetMUBets to determine if the bet was placed successfully. |
| FROM_COUNTRY_FORBIDDEN | Update request from restricted country |
| INVALID_MARKET_ID | Not used. |
| INVALID_NUMBER_OF_BETS | Number of bets not between 0 and 15 |
| MARKET_STATUS_INVALID | The status of the Market is invalid for this action. The Market may be suspended or closed. |
| MARKET_TYPE_NOT_SUPPORTED | The MarketID supplied refers to a market that is not supported by the API. |

Table 164: UpdateBetsResultEnum

| Value | Condition |
|-----------------------------------|--|
| OK | |
| BBM_DAEMON_NOT_AVAILABLE | The exchange encountered a critical error and was not able to match your bet. Your bet was not placed. |
| BET_CANCELLED_REMAINING_UNMATCHED | Bet partially cancelled, remainder is unmatched |
| BET_CANNOT_BE_ACCEPTED | Bet size cannot be accepted |
| BET_IN_PROGRESS | The exchange has not completed matching the bet before the bet matching time expired. You should |

| Value | Condition |
|---|---|
| BET_NOT_CANCELLED | use GetMUBets to determine if the bet was placed successfully. Bet could not be cancelled. This may be because the user's local wallet is suspended (for example, because Betfair have not yet received confirmation of your name and address). Part of the process of updating a bet involves cancellation of the original bet. However, if between the original bet and the UpdateBets request the local wallet that was used to fund the bet has been suspended, then the original bet will not be cancelled and the bet update will therefore not be successful. |
| BET_TAKEN_OR_LAPSED | Bet already matched or lapsed |
| BSP_BETS_CANNOT_BE_CANCELLED | You are attempting to cancel a BSP bet. |
| BSP_BETTING_NOT_ALLOWED | You are not allowed to modify a BSP bet |
| BSP_BETTING_NOT_SUPPORTED | This is not a BSP-enabled market |
| BSP_INVALID_PERSISTENCE | The new persistence must be one of BetPersistenceTypeEnum |
| BSP_INVALID_PRICE_LIMIT | The bet is a back bet and the newPrice is higher than the original or the bet is a lay bet and the newPrice is lower than the original. |
| BSP_MOC_BETS_EDITING_NOT_PERMITTED | You cannot change a BSP Market on Close bet |
| CANCELLED_NOT_PLACED | An internal error has occurred. Bet Cancelled but not placed. |
| ERROR_LINE_EXPO_BET_CANCELLED_NOT_PLACED | Bet already matched or lapsed |
| EVENT_CLOSED_CANNOT_MODIFY_BET | Event has been closed |
| EXCEEDED_EXPOSURE_OR_AVAILABLE_TO_BET_BALANCE | Bet exceeds exposure |
| EXPOSURE_CALCULATION_ERROR | Internal error calculating exposure |
| EXPOSURE_CALCULATION_IN_PROGRESS | The bet was not updated because the system is calculating your exposure. This error is most likely to occur when the account is close to the exposure limit or the available to bet balance is near zero, and you are placing/updating multiple bets in a short space of time. |
| FROM_COUNTRY_ON_EVENT_FORBIDDEN | Bet placed on restricted event from restricted country |
| INPLAY_FROM_COUNTRY_FORBIDDEN | Bet placed on in play event from restricted country |
| INSUFFICIENT_BALANCE | Bet exceeds available balance |
| INVALID_BET_ID | Invalid Bet ID supplied. One reason you might receive this message is that you have sent the request to the wrong exchange server. The UK exchange server cannot return information about bets placed on the Australian exchange server, and vice versa. |

| Value | Condition |
|----------------------------------|--|
| INVALID_BSP_BET_UPDATE | You have not submitted a new price for a BSP Limit on Close bet, or the new price is better than the existing price. |
| INVALID_INPUTS | A required field in the request was not supplied. For an exchange bet, you must supply the old and new price and size. |
| INVALID_NEW_PRICE_LIMIT | The price limit in the request must be lower for a back bet, or higher for a lay bet. |
| INVALID_OLD_PRICE | Invalid Odds supplied on original bet |
| INVALID_OLD_SIZE | Invalid stake supplied on original bet |
| INVALID_PRICE | Invalid Odds supplied for new bet |
| INVALID_PRICE_AND_SIZE | Invalid odds and stake supplied |
| INVALID_SIZE | Invalid stake supplied for new bet |
| LOSS_LIMIT_EXCEEDED | The potential loss for the updated bet exceeds the loss limit for the account |
| NEW_BET_SUBMITTED_SUCCESSFULLY | Bet edited successfully |
| NOT_PLACED_EXPOSURE_EXCEEDED | Internal Error |
| NOT_PLACED_REMAINING_CANCELLED | The new bet could not be placed but the remaining amounts were cancelled. |
| OK_REMAINING_CANCELLED | The remaining stake was cancelled and the new bet was submitted successfully |
| PARTIAL_CANCELLATION | Bet partially cancelled |
| PERSISTENCE_MODIFICATION_SUCCESS | The bet persistence has been changed |
| REMAINING_SIZE_CANCELLED | Remaining portion of bet cancelled |
| RUNNER_REMOVED | The runner has been removed from the event |
| UNKNOWN_ERROR | Internal Error |
| VACANT_TRAP | The selected trap is vacant |
| WRONG_MINIMUM_PERMITTED_BET_SIZE | Bet below minimum bet set |

**Note:**

If resultCode is not null then all bet edits have failed.

If resultCode is null then the size of betResults is equal to size of bets.

**Note:**

Betting on in-play event from Australia is forbidden.

**Note:**

Betting from Holland on events other than horse racing is forbidden.

Account Management API Services Reference

Add Payment Card (global)

The API AddPaymentCard service allows you to register a debit/credit card that you want to use for the purpose of depositing funds into your Betfair account, or for the purpose of withdrawing funds from your Betfair account.

Service Details

SOAP Action: addPaymentCard

Input

The following table describes the parameters used for calling the AddPaymentCard service.

Table 165: 1 instance of addPaymentCardReq

| Parameter | Mandatory? | Type | Description |
|------------|------------|---------------------------------------|--|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| cardNumber | Y | String | The account number of the card |
| cardType | Y | CardTypeEnum | The type of card (Visa, Mastercard, etc.) |
| cardStatus | Y | PaymentCardStatusEnum | The card's status. It can be either locked or unlocked (see Chapter 61). This parameter is mandatory and is required by Betfair for its internal procedures. It doesn't matter whether you specify LOCKED or UNLOCKED because the value you submit will be ignored. However, you must specify one or the other for the XML input to be valid. |

| Parameter | Mandatory? | Type | Description |
|-------------|------------|--------|---|
| startDate | N | String | The start date of the card, if applicable for the type of card |
| expiryDate | Y | String | The expiry date of the card |
| issueNumber | N | String | The card's issue number. For all cards except Switch and Solo cards, the value needs to be NULL (this is why the parameter is a string and not an integer). |
| billingName | Y | String | The name of the person listed as the account billing name |
| nickName | Y | String | An arbitrary name to use for referring to this card. The name must be 8 characters or less. |
| password | Y | String | The Betfair account login password |
| address1 | Y | String | The first line of the billing address for the card account |
| address2 | N | String | The second line of the billing address for the card account |
| address3 | N | String | The third line of the billing address for the card account |
| address4 | N | String | The fourth line of the billing address for the card account |
| town | Y | String | The town in which the cardholder resides. |
| county | Y | String | The county in which the cardholder resides. |
| zipCode | Y | String | The postal code of the billing address for the card account |
| country | Y | String | The country of the billing address for the card account |

Output

The following table describes the parameters returned from the addPaymentCard service.

Table 166: 1 Instance of addPaymentCardResp

| Parameter | Type | Description |
|----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| errorCode | see ... | Service specific error message. See below. |
| minorErrorCode | String | Reserved for future use - currently always nulls |
| paymentCard | PaymentCard | Details of the registered payment card |

The addPaymentCard service returns PaymentCard. The table below shows the fields in PaymentCard

Table 167: 1 instance of PaymentCard

| Parameter | Type | Description |
|--------------------|------------------------------|--|
| nickName | String | An arbitrary name to use for referring to this card |
| cardShortNumber | String | Partial card number |
| expiryDate | String | The expiry date of the card, in the form of MMY |
| startDate | String | The start date of the card, if applicable for the type of card, in the form of MMY |
| issueNumber | String | The card's issue number. For all cards except Switch and Solo cards, the value is NULL (this is why the parameter is a string and not an integer). |
| cardType | CardTypeEnum | The type of card (Visa, Mastercard, etc.) |
| issuingCountryIso3 | String | The country where the card was issued |
| totalDeposits | Double | The total amount deposited into the user's Betfair account from this card |
| totalWithdrawals | Double | The total amount withdrawn to this card from the user's Betfair account |
| netDeposits | Double | The total deposited minus the total withdrawn |

| Parameter | Type | Description |
|--------------------|--------|--|
| validationStatus | String | Indicates if the card has been validated by Betfair |
| billingName | String | The card billing name (the cardholder's name as it appears on the card) |
| billingAddress1 | String | The card billing address line one |
| billingAddress2 | String | The card billing address line two |
| billingAddress3 | String | The card billing address line three |
| billingAddress4 | String | The card billing address line four |
| town | String | The card billing address town |
| county | String | The card billing address county |
| postcode | String | The card billing address post code |
| billingCountryIso3 | String | The ISO3 country code for the card billing address |
| cardStatus | | The status of the card (LOCKED or UNLOCKED) indicating whether the card can be used to deposit or withdraw funds. If the card is locked, the betfair helpdesk should be contacted. |

Error Codes

The following table describes the error codes returned by addPaymentCard.

Table 168: AddPaymentCardErrorEnum

| Value | Condition |
|-------------------------------------|---|
| OK | |
| ACCOUNT_INACTIVE | The Betfair account is inactive |
| AGE_VERIFICATION_REQUIRED | The user cannot add a card because the user's age has not been verified |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| CARD_ALREADY_EXISTS | The card has already been registered to this account |
| CARD_NOT_VALID_FOR_ACCOUNT_CURRENCY | The card's currency does not match the account currency |
| INVALID_BILLING_NAME | The card billing account name does not match the Betfair account name |

| Value | Condition |
|---------------------------------|---|
| INVALID_CARD_ADDRESS | The string did not appear to be an address |
| INVALID_CARD_CV2 | Not used |
| INVALID_CARD_DETAILS | The card nickname is not valid |
| INVALID_CARD_NUMBER | The card number does not appear valid |
| INVALID_CARD_TYPE | The card type is not supported. For a list of the supported cards, see Table 61-11 on page 216. |
| INVALID_COUNTRY_CODE | The country code does not appear valid |
| INVALID_EXPIRY_DATE | The expiry date is invalid or too soon |
| INVALID_ISSUE_NUMBER | The issue number submitted is not valid for type of card specified. The issue number for that type of card needs to be a two-digit integer. |
| INVALID_PASSWORD | The password is not valid |
| INVALID_START_DATE | The start date does not appear valid |
| INVALID_ZIP_CODE | The postal code does not appear valid |
| MAXIMUM_NUMBER_OF_CARDS_REACHED | The account already has the maximum number of payment cards allowed |
| NOT_FUNDED_WITH_FIRST_CARD | The account has registered a previous card, but no money has been deposited to the account with that card |
| UNAUTHORIZED | The account is not authorised to add a payment card |

Delete Payment Card (global)

The API DeletePaymentCard service allows you to remove a previously registered payment card from an account.

Service Details

SOAP Action: deletePaymentCard

Input

The following table describes the parameters used for calling the DeletePaymentCard service.

Table 169: 1 instance of DeletePaymentCardReq

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| nickName | Y | String | An arbitrary name used for referring to this card |

| Parameter | Mandatory? | Type | Description |
|-----------|------------|--------|------------------------------------|
| password | Y | String | The Betfair account login password |

Output

The following table describes the parameters returned from the AddPaymentCard service.

Table 170: 1 Instance of DeletePaymentCardResp

| Parameter | Type | Description |
|-----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| errorCode | see ... | Service specific error message. See below. |
| minorErrorCode | String | Reserved for future use - currently always nulls |
| nickName | String | An arbitrary name to use for referring to this card |
| billingName | String | The name of the person listed as the account billing name |
| cardShortNumber | String | Partial card number |
| cardType | CardTypeEnum | The type of card (Visa, Mastercard, etc.) |
| issuingCountry | String | The country where the card was issued |
| expiryDate | String | The expiry date of the card, in the form of MYY |

Error Codes

The following table describes the error codes returned by DeletePaymentCard.

Table 171: DeletePaymentCardErrorEnum

| Value | Condition |
|----------------------|---|
| OK | |
| ACCOUNT_INACTIVE | The account is locked or suspended |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| CARD_NOT_DELETED | The card was not removed due to an unspecified error |
| INVALID_CARD_DETAILS | The card nickname was not recognized |
| INVALID_PASSWORD | The account password was incorrect |
| UNAUTHORIZED | The account is not authorised to manage payment cards |

Deposit From Payment Card (global)

The API `DepositFromPaymentCard` service allows you to deposit funds into your UK wallet from a previously registered payment card. You cannot deposit funds directly into your Australian wallet. The `transferFunds` service is provided to enable you to move funds into your Australian wallet from your UK wallet (see [Chapter 54](#)).

Service Details

SOAP Action: `depositFromPaymentCard`

Input

The following table describes the parameters used for calling the `DepositFromPaymentCard` service.

Table 172: 1 instance of `DepositFromPaymentCardReq`

| Parameter | Mandatory? | Type | Description |
|-----------------------------|------------|----------------------------------|---|
| <code>header</code> | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| <code>amount</code> | Y | Double | The amount to deposit from the payment card |
| <code>cardIdentifier</code> | Y | String | The nickname for the payment card |
| <code>cv2</code> | Y | String | The CV2 digits from the card |
| <code>password</code> | Y | String | The account password |

Output

The following table describes the parameters returned from the `DepositFromPaymentCard` service.

Table 173: 1 Instance of `DepositFromPaymentCardResp`

| Parameter | Type | Description |
|------------------------|-----------------------------------|--|
| <code>header</code> | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| <code>errorCode</code> | see ... | Service specific error codes. See below. |
| <code>fee</code> | Double | The card transaction fee for this deposit |
| <code>maxAmount</code> | Double | The maximum deposit amount allowed for this card. If <code>errorCode</code> is set to <code>CARD_AMOUNT_OUTSIDE_LIMIT</code> |

| Parameter | Type | Description |
|----------------|--------|---|
| minAmount | Double | then minAmount and maxAmount is set. If errorCode is set to DEPOSIT_LIMIT_EXCEEDED then maxAmount is set. The minimum deposit amount allowed for this card. If errorCode is set to CARD_AMOUNT_OUTSIDE_LIMIT then minAmount and maxAmount is set. If errorCode is set to DEPOSIT_LIMIT_EXCEEDED then maxAmount is set. |
| minorErrorCode | String | This field contains age.verification.required if the user has not completed the age verification process with Betfair. |
| netAmount | Double | The amount deposited to the Betfair account less any fee. |
| transactionId | String | A unique ID for this deposit. |

Error Codes

The following table describes the error codes returned by DepositFromPaymentCard. The PaymentsErrorEnum table applies to both deposits and withdrawals so not all enums apply to deposits.

Table 174: PaymentsErrorEnum

| Value | Condition |
|---------------------------|---|
| OK | |
| ACCOUNT_SUSPENDED | The Betfair account is currently suspended |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| CARD_AMOUNT_OUTSIDE_LIMIT | The deposit amount is lower or higher than the allowed amount for the card |
| CARD_EXPIRED | The card has an expiration date in the past |
| CARD_LOCKED | The card is locked for security reasons |
| CARD_NOT_FOUND | The card is not registered |
| CFT_MAX_WITHDRAWAL_LIMIT | The amount available to withdraw has been exceeded |
| DEPOSIT_DECLINED | The deposit was declined |
| DEPOSIT_LIMIT_EXCEEDED | The deposit exceeded the GamCare deposit limit amount |
| EXCEEDS_BALANCE | The deposit exceeds the amount available on the card |

| Value | Condition |
|-------------------------------|---|
| INVALID_AMOUNT | The amount entered is not a valid figure |
| INVALID_CARD_CV2 | The CV2 digits are not valid |
| INVALID_CARD_DETAILS | The card was not recognised |
| INVALID_EXPIRY_DATE | The expiration date was not a valid date |
| INVALID_MASTERCARD | Withdrawal to a non-UK Mastercard not allowed until the card is validated by Betfair |
| INVALID_PASSWORD | The account password is invalid |
| NEGATIVE_NET_DEPOSITS | The amount to withdraw is more than the amount deposited from a credit card |
| NON_STERLING_TO_UK_MASTERCARD | The currency of the account is not in Sterling and the card is a UK sterling Mastercard. |
| NON_ZERO_NON_NEG_NET_DEPOSITS | The account has funds deposited from other cards. You must withdraw the amount deposited from the other card(s) before you can withdraw to this card. |
| UNAUTHORIZED | The account is not authorised to manage payment cards |
| VISA_WITHDRAWAL_NOT_POSSIBLE | The VISA card policy does not allow withdrawal to this card. |

Forgot Password (global)

The API ForgotPassword service allows you to reset the password on an account. When an account is created, two security questions and answers are created for the account. The forgotPassword service changes the password to the new one provided when the correct answers to the security questions are included.

You normally use the forgotPassword service by calling it twice. The first call is used to retrieve the security questions associated with the Betfair account. You then call forgotPassword a second time, supplying the answers to the security questions and the new password.

The first time you call forgotPassword, you send the following parameters, in addition to the header:

- username
- emailAddress
- countryOfResidence

The response contains security questions that you should present to the user. You then call forgotPassword a second time with the following parameters, in addition to the header:

- username
- emailAddress
- countryOfResidence
- forgottenPasswordAnswer1
- forgottenPasswordAnswer2
- newPassword
- newPasswordRepeat

Note:



You can make a single call to `forgotPassword` if you already know the correct answers to the security questions. However, Betfair highly recommends that you not store the security questions and answers in your application and, instead, use the `forgotPassword` service to retrieve them when needed.

Service Details

SOAP Action: `forgotPassword`

Input

The following table describes the parameters used for calling the `forgotPassword` service.

Table 175: 1 instance of `ForgotPasswordReq`

| Parameter | Mandatory? | Type | Description |
|---------------------------------------|------------|----------------------------------|---|
| <code>header</code> | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| <code>username</code> | Y | String | The username of the account |
| <code>emailAddress</code> | Y | String | The e-mail address associated with the account |
| <code>countryOfResidence</code> | Y | String | The country associated with the account |
| <code>forgottenPasswordAnswer1</code> | N | String | The answer to the first security question |
| <code>forgottenPasswordAnswer2</code> | N | String | The answer to the second security question |
| <code>newPassword</code> | N | String | The new password for the account |
| <code>newPasswordRepeat</code> | N | String | The new password for the account |

Output

The following table describes the parameters returned from the `forgotPassword` service.

Table 176: 1 Instance of `ForgotPasswordResp`

| Parameter | Type | Description |
|------------------------|--------------------------------------|---|
| <code>header</code> | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| <code>errorCode</code> | <code>ForgotPasswordErrorEnum</code> | Service specific error code. See below. |

| Parameter | Type | Description |
|----------------|--------|--|
| minorErrorCode | String | Reserved for future use. Currently always nil. |
| question1 | String | The first security question associated with the account |
| question2 | String | The second security question associated with the account |

Error Codes

The following table describes the error codes returned by forgotPassword.

Table 177: ForgotPasswordErrorEnum

| Value | Condition |
|-------------------------------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_ANSWER | The security answers were not correct |
| INVALID_COUNTRY_OF_RESIDENCE | The country of residence was not correct |
| INVALID_EMAIL | The e-mail address was not correct |
| INVALID_PASSWORD | The password was not correct |
| INVALID_USERNAME | The username was not recognised |
| TOO_MANY_ATTEMPTS_ACCOUNT_SUSPENDED | The account has been suspended after too many incorrect attempts |

Get Account Funds (exchange)

The API GetAccountFunds service allows you to retrieve information about your local wallet on a particular exchange server. For an explanation of the concept of wallets, see [“Using Region-specific Wallets for Placing Bets” on page 12](#).

Service Details

SOAP Action: getAccountFunds



Note:

The GetAccountFunds service is provided as an exchange service (in other words, it is available from the exchange servers) and not as a global service. The service provides information about the funds in your local wallet only. If you send a request for the service to the UK exchange server, the response tells you the balance of funds in your UK wallet only. If you send it to the Australian server, the response tells you about the funds in your Australian wallet only.

Input

The following table describes the parameters used for calling the GetAccountFunds service.

Table 178: 1 instance of AccountFundsReq

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |

Output

The following table describes the parameters returned from the GetAccountFunds service.

Table 179: 1 Instance of AccountFundsResp

| Parameter | Type | Description |
|----------------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| availBalance | Decimal | Current balance less exposure and retained commission |
| balance | Decimal | Current balance |
| commissionRetain | Decimal | Commission potentially due on markets which have not been fully settled |
| creditLimit | Decimal | Amount of credit available |
| currentBetfairPoints | Integer | Total of Betfair Points awarded based on commissions or implied commissions paid. (This parameter will not be populated for the Australian wallet.) |
| expoLimit | Decimal | Total exposure allowed |
| exposure | Decimal | Returned as a negative figure. Total funds tied up with current bets. |
| holidaysAvail | Integer | Betfair Holidays to be used to prevent the weekly decay of Betfair Points. There is a maximum of 4. (This parameter will not be populated for the Australian wallet.) |
| minorErrorCode | String | Reserved for future use - currently always null |
| nextDiscount | Decimal | Discount to be applied when commission is next calculated. (This parameter will not be |

| Parameter | Type | Description |
|-----------------|--------------------------|---|
| withdrawBalance | Decimal | populated for the Australian wallet.) Balance available for withdrawal |
| errorCode | GetAccountFundsErrorEnum | Specific error code GetAccountFundsErrorEnum |

Error Codes

The following table describes the error codes returned by `getAccountFunds`.

Table 180: GetAccountFundsErrorEnum

| Value | Condition |
|---------------------------|--|
| OK | |
| EXPOSURE_CALC_IN_PROGRESS | The API is busy calculating your current exposure. |
| API_ERROR | General API error. Check the APIResponseHeader <code>errorCode</code> field for additional information |

Get Account Statement (exchange)

The API `GetAccountStatement` service allows you to obtain information about transactions involving your local wallet on an exchange server.



Note:

This service returns transactions involving a single wallet only. If you send your request to the UK exchange server, it returns a statement of transactions involving only your UK wallet. If you send it to the Australian exchange server, the response concerns only your Australian wallet. Remember that withdrawals and deposits can only be made from and into the UK wallet. (To transfer funds into and out of your Australian wallet you must use the `TransferFunds` service) Therefore, requests to the Australian exchange server will contain no details of deposits and withdrawals.

Service Details

SOAP Action: `getAccountStatement`



Note:

The `GetAccountStatement` service is provided as an exchange service (in other words, it is available from the exchange server) instead of as a global service.

Known Issues

If `itemsIncluded = EXCHANGE`, the `startdate` and `enddate` properties on the `GetAccountStatementReq` are ignored. `startDate` and `endDate` are used when `itemsIncluded = ALL`

Input

The following table describes the parameters used for calling the `GetAccountStatement` service.

Table 181: 1 instance of `GetAccountStatementReq`

| Parameter | Mandatory? | Type | Description |
|---------------|------------|---|--|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| startRecord | Y | Integer | The first record number to return (supports paging). Record numbering starts from 0. For example, to retrieve the third record and higher, set startRecord to 2. |
| recordCount | Y | Int | The maximum number of records to return |
| startDate | Y | Date | Return records on or after this date |
| endDate | Y | Date | Return records on or before this date |
| itemsIncluded | Y | AccountStatementIncludeEnum | Determines what type of statements items to return. For more information, see Chapter 61 . |
| locale | N | String | Specify the language for the marketName reply if you want a different language than the account default. For more information, |

Output

The following table describes the parameters returned from the `GetAccountStatement` service.

Table 182: 1 Instance of `GetAccountStatementResp`

| Parameter | Type | Description |
|-----------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| items | ArrayOfAccountStatementItem | Market type (see below) |

| Parameter | Type | Description |
|------------------|---------------------------|---|
| errorCode | AccountStatementErrorEnum | Specific error code (see enum below) |
| minorErrorCode | String | Reserved for future use - currently always null |
| totalRecordCount | Integer | Total number of records matching the selection criteria |

The GetAccountStatement service returns items as an array of AccountStatementItem. The table below shows the fields in AccountStatementItem

Table 183: ArrayOfAccountStatementItem 0 or more instances of AccountStatementItem

| Parameter | Type | Description |
|-----------------|-------------------------------------|--|
| accountBalance | Double | Account Balance |
| Amount | Double | The amount won / lost for bets or amount deposited / withdrawn in the account currency |
| avgPrice | Double | The average matched price of the bet (null if no part has been matched) |
| betId | Long | Unique identifier generated for every bet placement |
| betSize | Double | The amount of the stake of your bet. (0 for commission payments or deposit/ withdrawals) |
| betType | BetTypeEnum | Back or lay - please see Chapter 61 for details. |
| betCategoryType | BetCategoryTypeEnum | Exchange, Market on Close SP bet, or Limit on Close SP bet. |
| commissionRate | String | Commission rate on market |
| eventId | Integer | Id of the market |
| eventType | Integer | Event Type |
| fullMarketName | String | Full Market Name. For card payment items, this field contains the card name |
| grossBetAmount | String | Gross Bet Amount |
| marketName | String | Market Name. For card transactions, this field indicates the type of card transaction (deposit, deposit fee, or withdrawal). |
| marketType | MarketTypeEnum | Market type - please see Chapter 61 . For account deposits and withdrawals, |

| Parameter | Type | Description |
|-----------------|--------------------------------------|---|
| | | marketType is set to NOT_APPLICABLE. |
| placedDate | Long | Date and time of bet placement |
| selectionId | Integer | Id of the selection (this will be the same for the same selection across markets) |
| selectionName | Integer | Name of the selection |
| settledDate | Long | Date and time at the bet portion was settled |
| startDate | Long | Start date of the market |
| transactionType | AccountStatementEnum | Debit or credit - please see Chapter 61 for details. |
| winLose | AccountStatementEnum | Win or loss - please see Chapter 61 for details. |
| executedBy | String | Betfair Internal. This field is always set to UNKNOWN |

Error Codes

The following table describes the error codes returned by GetAccountStatement.

Table 184: GetAccountStatementErrorEnum

| Value | Condition |
|--------------------------------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_END_DATE | End date is not supplied or is invalid |
| INVALID_LOCALE_DEFAULTING_TO_ENGLISH | The locale string was not recognized. Returned results are in English. |
| INVALID_RECORD_COUNT | Max Records < 0 or > 100 |
| INVALID_START_DATE | Start date is not supplied or is invalid |
| INVALID_START_RECORD | Start record is not supplied or is invalid |
| NO_RESULTS | No transactions meet the specified criteria |

Get Payment Card (global)

The API GetPaymentCard service allows you to retrieve the details of your registered payment cards. You can use this call to determine what cards have been registered, the balance of funds deposited and available to withdraw, and the nickname the card was registered with.

Service Details

SOAP Action: `getPaymentCard`

Input

The following table describes the parameters used for calling the `getPaymentCard` service.

Table 185: 1 instance of `GetPaymentCardReq`

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |

Output

The following table describes the parameters returned from the `getPaymentCard` service.

Table 186: 1 Instance of `GetPaymentCardResp`

| Parameter | Type | Description |
|------------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| errorCode | see ... | Service specific error message. See below. |
| minorErrorCode | String | Reserved for future use - currently always nulls |
| paymentCardItems | ArrayOfPaymentCard | An array of type <code>PaymentCard</code> containing details of registered cards. |

The `getPaymentCard` service returns `PaymentCard`. The table below shows the fields in `PaymentCard`

Table 187: 1 instance of `PaymentCard`

| Parameter | Type | Description |
|-----------------|------------------------------|--|
| nickName | String | An arbitrary name to use for referring to this card |
| cardShortNumber | String | Partial card number |
| expiryDate | String | The expiry date of the card, in the form of MMY |
| startDate | String | The start date of the card, if applicable for the type of card, in the form of MMY |
| issueNumber | String | The issue number, if applicable for the type of card |
| cardType | CardTypeEnum | The type of card (Visa, Mastercard, etc.) |

| Parameter | Type | Description |
|--------------------|---------------------------------------|---|
| issuingCountryIso3 | String | The country where the card was issued |
| totalDeposits | Double | The total amount deposited into the user's Betfair account from this card |
| totalWithdrawals | Double | The total amount withdrawn to this card from the user's Betfair account |
| netDeposits | Double | The total deposited minus the total withdrawn |
| validationStatus | String | Indicates if the card has been validated by Betfair |
| billingAddress1 | String | The card billing address line one |
| billingAddress2 | String | The card billing address line two |
| billingAddress3 | String | The card billing address line three |
| billingAddress4 | String | The card billing address line four |
| town | String | The card billing address town |
| postcode | String | The card billing address post code |
| billingCountryIso3 | String | The ISO3 country code for the card billing address |
| cardStatus | PaymentCardStatusEnum | The status of the card (LOCKED or UNLOCKED), indicating whether the card can be used to deposit or withdraw funds. If the card is locked, contact customer support. |

Error Codes

The following table describes the error codes returned by `GetPaymentCard`.

Table 188: `GetPaymentCardErrorEnum`

| Value | Condition |
|------------------|--|
| OK | |
| ACCOUNT_INACTIVE | The account is suspended or locked |
| API_ERROR | General API error. Check the APIResponseHeader <code>errorCode</code> field for additional information |
| INVALID_PASSWORD | The password is not valid |
| UNAUTHORIZED | The account is not authorised to manage payment cards |

Get Subscription Info (global)

The API GetSubscriptionInfo service returns information on your API subscription.

Service Details

SOAP Action: `getSubscriptionInfo`

Input

The following table describes the parameters used for calling the GetSubscriptionInfo service.

Table 189: 1 instance of GetSubscriptionInfoReq

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |

Output

The following table describes the parameters returned from the GetSubscriptionInfo service.

Table 190: 1 Instance of GetSubscriptionInfoResp

| Parameter | Type | Description |
|----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| minorErrorCode | String | Reserved for future use - currently always null |
| subscription | ArrayOfSubscription | Subscription information |

The GetSubscriptionInfo service returns subscriptions as an array of Subscription. The table below shows the fields in Subscription

Table 191: ArrayOfSubscription 0 or more instances of Subscription

| Parameter | Type | Description |
|---------------|-----------------------------------|--|
| billingAmount | Decimal | Subscription payment amount |
| billingDate | dateTime | Next billing date |
| billingPeriod | BillingPeriodEnum | Billing period - please see Chapter 61 . |
| productId | Integer | Product ID |
| productName | String | The name of the subscription product |

| Parameter | Type | Description |
|-------------------|--|--|
| services | ArrayOfServiceCall | Information on the subscription services |
| setupChargeActive | Boolean | If True there will be a setup charge |
| setupCharge | Decimal | Amount of setup charge |
| status | SubscriptionStatusEnum | Status of subscription - please see Chapter 61 . |
| subscribedDate | dateTime | The date the subscription was enabled |
| vatEnabled | Boolean | If True, VAT will be added on top of the Billing Amount |

The Subscription item returns `services` as an array of `ServiceCall`. The table below shows the fields in Subscription

Table 192: ArrayOfServiceCall 0 or more instances of ServiceCall

| Parameter | Type | Description |
|--------------|-----------------------------|--|
| maxUsages | Integer | Throttle usage amount |
| period | Long | Throttle limit time |
| periodExpiry | dateTime | Throttle expiration date |
| serviceType | ServiceEnum | Services available in the subscription. For more information, see Chapter 61 . |
| usageCount | Integer | Current usage count |

Modify Password (global)

The API ModifyPassword service allows you to change the password associated with the current login account.

Service Details

SOAP Action: `modifyPassword`

Input

The following table describes the parameters used for calling the modifyPassword service.

Table 193: 1 instance of ModifyPasswordReq

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |

| Parameter | Mandatory? | Type | Description |
|-------------------|------------|--------|---|
| password | Y | String | The current password for the account |
| newPassword | Y | String | The desired new password for the account. This password must conform to the same password rules as the Betfair Website. |
| newPasswordRepeat | Y | String | The desired new password for the account. It is recommended that you force the user to enter the password twice. |

Output

The following table describes the parameters returned from the modifyPassword service.

Table 194: 1 Instance of ModifyPasswordResp

| Parameter | Type | Description |
|----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| errorCode | see ... | Service specific error codes. See below. |
| minorErrorCode | String | Reserved for future use. Currently always nil. |

Error Codes

The following table describes the error codes returned by modifyPassword.

Table 195: ModifyPasswordErrorEnum

| Value | Condition |
|----------------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_NEW_PASSWORD | The new password supplied is not valid |
| INVALID_PASSWORD | The existing password supplied is not correct |
| PASSWORDS_DONT_MATCH | The supplied new passwords do not match each other. |

Modify Profile (global)

The API ModifyProfile service allows you to change the address, telephone numbers, and gamcare limits for an account.

Service Details

SOAP Action: `modifyProfile`

Input

The following table describes the parameters used for calling the modifyProfile service.

Table 196: 1 instance of ModifyProfileReq

| Parameter | Mandatory? | Type | Description |
|-------------|------------|----------------------------------|--|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| password | Y | String | The account password |
| address1 | N | String | The first line of the account address |
| address2 | N | String | The second line of the account address |
| address3 | N | String | The third line of the account address |
| townCity | N | String | The town or city of the account address |
| countyState | N | String | The county or state of the account address. Note that, for customers whose <code>countryOfResidence</code> is Australia, the <code>countyState</code> field must specify the state (not the county) of residence. Valid values are: Australian Capital Territory, New South Wales, Victoria, Queensland, South Australia, Western Australia, Tasmania. |
| postCode | N | String | The postal code of the account address |

| Parameter | Mandatory? | Type | Description |
|-----------------------|------------|--------------------------------------|--|
| countryOfResidence | N | String | The country of the account address |
| homeTelephone | N | String | The home telephone number of the account holder |
| workTelephone | N | String | The work telephone number of the account holder |
| mobileTelephone | N | String | The mobile telephone number of the account holder |
| emailAddress | N | String | The registered e-mail address for the account |
| timeZone | N | String | The account holder's time zone |
| depositLimit | N | Integer | The Gamcare deposit limit for the account. (When you set this value, it applies to both the UK wallet and the Australian wallet.) |
| depositLimitFrequency | N | GamcareLimitFreqEnum | Whether the deposit limit is to be applied per day, per week, per month, or per year. (When you set this value, it applies to both the UK wallet and the Australian wallet.) |
| lossLimit | N | Integer | The Gamcare loss limit for the account. (When you set this value, it applies to both the UK wallet and the Australian wallet.) |
| lossLimitFrequency | N | GamcareLimitFreqEnum | Whether the loss limit is to be applied per day, per week, per month, or per year. (When you set this value, it applies to both the UK wallet and the Australian wallet.) |

| Parameter | Mandatory? | Type | Description |
|--------------------|------------|--------|--|
| nationalIdentifier | N | String | If requestVersion contains V1, this field is returned containing the national identifier set for the user's account profile. This is currently used only by Betfair to store the national identity number for Indian and South African customer accounts. See: http://en.wikipedia.org/wiki/Permanent_account_number and http://en.wikipedia.org/wiki/National_identification_number#South African |

Output

The following table describes the parameters returned from the modifyProfile service.

Table 197: 1 Instance of ModifyProfileResp

| Parameter | Type | Description |
|------------------|---------------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| errorCode | ModiyProfileErrorEnum | Service specific error code. See below. |
| minorErrorCode | String | Reserved for future use. Currently always nil. |
| validationErrors | ValidationErrorsEnum | An array containing validation errors for the submitted profile modifications |

Error Codes

The following table describes the error codes returned by modifyProfile.

Table 198: ModifyProfileErrorEnum

| Value | Condition |
|-------------------|--|
| OK | |
| VALIDATION_ERRORS | There was a problem with the submitted information. Check the validationErrors array for details of the specific problem. For Australian |

| Value | Condition |
|----------------------------|--|
| | customers, check that the countyState parameter has been correctly specified in the ModifyProfile request. |
| PROFILE_MODIFICATION_ERROR | There was problem encountered when trying to modify the profile |
| UNAUTHORIZED | The user is not allowed to modify the account profile |
| INVALID_PASSWORD | The submitted password is not valid |
| ACCOUNT_INACTIVE | The account is either suspended or locked |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |

Retrieve LIMB Message (global)

The API RetrieveLIMBMessage service allows to get messages from Betfair for the customer. Log In Message Board (LIMB) messages are presented when the customer logs in and often require the customer to respond before a certain date. If the customer does not respond within a certain amount of time to some of the messages, the customer's account may be suspended. You should present the message to the customer and then submit the customer's responses using the service.

If the login account is suspended and the call to Login fails, you can still call retrieveLIMBMessage and submitLIMBMessage.

Service Details

SOAP Action: `retrieveLIMBMessage`

Input

The following table describes the parameters used for calling the retrieveLIMBMessage service.

Table 199: 1 instance of RetrieveLIMBMessageReq

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |

Output

The following table describes the parameters returned from the retrieveLIMBMessage service.

Table 200: 1 Instance of RetrieveLIMBMessageResp

| Parameter | Type | Description |
|-----------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |

| Parameter | Type | Description |
|--------------------------------------|-------------------------|--|
| errorCode | see ... | Service specific response code. See below |
| minorErrorCode | String | Reserved for future use. Currently always nil. |
| totalMessageCount | Integer | The total number of message blocks returned. |
| retrievePersonalMessage | see ... | Personal messages |
| retrieveTCPrivacyPolicyChangeMessage | see ... | Policy change messages |
| retrievePasswordChangeMessage | see ... | Password change messages |
| retrieveBirthDateCheckMessage | see ... | Birthdate confirmation |
| retrieveAddressCheckMessage | see ... | Address confirmation |
| retrieveContactDetailsCheckMessage | see ... | Contact details confirmation |
| retrieveChatNameChangeMessage | see ... | Chat name change request |
| retrieveCardBillingAddressCheck | CheckItems | Card address confirmation |

The retrieveLIMBMessage service returns a number of message blocks. The tables below shows the fields for each message type

Table 201: RetrievePersonalMessage

| Parameter | Type | Description |
|-------------|----------|---|
| messageID | Int | The ID of the message |
| enforceDate | DateTime | The date when the user must reply to the message |
| indicator | Boolean | If true, there is a message of this type and a response is required |
| message | String | The message |

Table 202: RetrieveTCPrivacyPolicyChangeMessage

| Parameter | Type | Description |
|-----------------|----------|---|
| messageID | Int | The ID of the message |
| enforceDate | DateTime | The date when the user must reply to the message |
| indicator | Boolean | If true, there is a message of this type and a response is required |
| reasonForChange | String | An explanation of the T and C change |

Table 203: RetrievePasswordChangeMessage

| Parameter | Type | Description |
|-------------|----------|---|
| messageID | Int | The ID of the message |
| enforceDate | DateTime | The date when the user must reply to the message |
| indicator | Boolean | If true, there is a message of this type and a response is required |

Table 204: RetrieveBirthDateCheckMessage

| Parameter | Type | Description |
|-------------|----------|---|
| messageID | Int | The ID of the message |
| enforceDate | DateTime | The date when the user must reply to the message |
| indicator | Boolean | If true, there is a message of this type and a response is required |
| birthDate | DateTime | The birth date currently held by Betfair |

Table 205: RetrieveAddressCheckMessage

| Parameter | Type | Description |
|-------------|----------|---|
| messageID | Int | The ID of the message |
| enforceDate | DateTime | The date when the user must reply to the message |
| indicator | Boolean | If true, there is a message of this type and a response is required |
| address1 | String | The first line of the address currently held by Betfair |
| address2 | String | The second line of the address currently held by Betfair |
| address3 | String | The third line of the address currently held by Betfair |
| town | String | The town of the address currently held by Betfair |
| county | String | The county of the address currently held by Betfair |
| zipCode | String | The postal code of the address currently held by Betfair |
| country | String | The country of the address currently held by Betfair |

Table 206: RetrieveContactDetailsCheckMessage

| Parameter | Type | Description |
|-----------------|----------|---|
| messageID | Int | The ID of the message |
| enforceDate | DateTime | The date when the user must reply to the message |
| indicator | Boolean | If true, there is a message of this type and a response is required |
| homeTelephone | String | The home telephone number currently held by Betfair |
| workTelephone | String | The work telephone number currently held by Betfair |
| mobileTelephone | String | The mobile telephone number currently held by Betfair |
| emailAddress | String | The e-mail address currently held by Betfair |

Table 207: RetrieveChatNameChangeMessage

| Parameter | Type | Description |
|-------------|----------|---|
| messageID | Int | The ID of the message |
| enforceDate | DateTime | The date when the user must reply to the message |
| indicator | Boolean | If true, there is a message of this type and a response is required |
| chatName | String | The customer's current chat name |

The retrieveLIMBMessage service can return an array of retrieveCardBillingAddressCheckLIMBMessage for each billing card.

Table 208: ArrayOfRetrieveCardBillingAddressCheckLIMBMessage

| Parameter | Type | Description |
|-----------------|----------|---|
| messageID | Int | The ID of the message |
| enforceDate | DateTime | The date when the user must reply to the message |
| indicator | Boolean | If true, there is a message of this type and a response is required |
| nickName | DateTime | The card nickname |
| cardShortNumber | String | The last four digits of the card |
| address1 | String | The card billing address line one |

| Parameter | Type | Description |
|-----------|--------|--------------------------------------|
| address2 | String | The card billing address line two |
| address3 | String | The card billing address line three |
| town | String | The card billing address town |
| county | String | The card billing address county |
| zipCode | String | The card billing address postal code |
| country | String | The card billing address country |

Error Codes

The following table describes the error codes returned by retrieveLIMBMessage.

Table 209: RetrieveLIMBMessageErrorEnum

| Value | Condition |
|-----------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |

Self Exclude (global)

The API SelfExclude service allows you to provide users with the ability to self-exclude from the Betfair sports exchange for six months. Once the request is successfully processed, the user's session is invalidated and further API calls (except Login) will fail with a NO_SESSION error. Calls to the Login service will fail with the message ACCOUNT_CLOSED

You should display the following text to your user to explain what happens if they self-exclude:

“Self-exclusion means that your Betfair account will remain closed for a minimum period of six months and will not be re-activated under **any circumstances** during the exclusion period. Once you self-exclude, you will not be able to cancel unmatched bets or withdraw money from your account, however, you can contact the Betfair helpdesk to cancel bets or withdraw money from your account.”

Service Details

SOAP Action: selfExclude

Input

The following table describes the parameters used for calling the selfExclude service.

Table 210: 1 instance of selfExcludeReq

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session |

| Parameter | Mandatory? | Type | Description |
|-------------|------------|---------|---|
| | | | token (see Betfair Simple Data Types). |
| selfExclude | Y | Boolean | This must be True |
| password | Y | String | The user's account password |

Output

The following table describes the parameters returned from the selfExclude service.

Table 211: 1 Instance of SelfExcludeResp

| Parameter | Type | Description |
|----------------|-----------------------------------|--|
| header | APIResponseHeader | The API Response Header |
| minorErrorCode | String | Reserved for future use. Currently always nil. |
| errorCode | see ... | Service specific error code (see below). |

Error Codes

The following table describes the error codes returned by selfExclude.

Table 212: SelfExcludeErrorEnum

| Value | Condition |
|----------------------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| ACCOUNT_CLOSED | The user's account is closed already |
| INVALID_PASSWORD | The password supplied is incorrect. |
| INVALID_SELF_EXCLUDE_VALUE | The value of selfExclude in the request must be true |

Set Chat Name(global)

The API SetChatName service allows you to set or change the user's chat (forum) name.

Service Details

SOAP Action: setChatName

Input

The following table describes the parameters used for calling the setChatName service.

Table 213: 1 instance of SetChatNameReq

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| chatName | Y | String | The desired chat name |
| password | Y | String | The user's password |

Output

The following table describes the parameters returned from the setChatName service.

Table 214: 1 Instance of SetChatNameResp

| Parameter | Type | Description |
|----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| errorCode | see ... | Service specific error code. See below. |
| minorErrorCode | String | Reserved for future use. Currently always nil. |

Error Codes

The following table describes the error codes returned by setChatName.

Table 215: SetChatNameErrorEnum

| Value | Condition |
|-------------------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| INVALID_PASSWORD | The password was not correct |
| ACCOUNT_SUSPENDED | The account is suspended |
| ACCOUNT_NOT_FUNDED | Chat names are only available to funded, active accounts. The account must have a positive balance in the UK wallet and at least one entry in the Account Statement table |
| CHAT_NAME_UNAVAILABLE | The chat name requested cannot be used or is already in use |
| CANNOT_CHANGE_CHAT_NAME | The user's chat name cannot be changed |

Submit LIMB Message (global)

The API SubmitLIMBMessage service allows you to send responses from the customer to messages sent from Betfair. You retrieve these messages using the service.

Service Details

SOAP Action: submitLIMBMessage

Input

The following table describes the parameters used for calling the submitLIMBMessage service.

Table 216: 1 instance of SubmitLIMBMessageReq

| Parameter | Mandatory? | Type | Description |
|------------------------------------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| password | Y | String | The customer's password |
| submitPersonalMessage | N | see ... | Personal message response |
| submitTCPrivacyPolicyChangeMessage | Y | see ... | Policy change response |
| submitPasswordChangeMessage | N | see ... | Password change |
| submitBirthDateCheckMessage | N | see ... | Birth date confirmation |
| submitAddressCheckMessage | N | see ... | Address confirmation |
| submitContactDetailsCheckMessage | N | see ... | Contact details confirmation |
| submitChatNameChangeMessage | N | see ... | Chat name change |
| submitCardBillingAddressCheckItems | N | see ... | Billing address confirmation |

You use the submitLIMBMessage service to send one or more message blocks. The tables below show the fields for each message type.

Table 217: SubmitPersonalMessage

| Parameter | Type | Description |
|-----------------|--------|---------------------------------------|
| messageID | Int | The ID of the message |
| acknowledgement | String | Submit 'Y' to acknowledge the message |

Table 218: SubmitTCPrivacyPolicyChangeMessage

| Parameter | Type | Description |
|---------------------------------|--------|--|
| messageID | Int | The ID of the message |
| tCPrivacyPolicyChangeAcceptance | String | Submit 'Y' to indicate acceptance of the policy change |

Table 219: SubmitPasswordChangeMessage

| Parameter | Type | Description |
|-------------------|---------|--------------------------------|
| messageID | Int | The ID of the message |
| newPassword | String | The new account password |
| newPasswordRepeat | Boolean | The new account password again |

Table 220: SubmitBirthDateCheckMessage

| Parameter | Type | Description |
|------------------|----------|---|
| messageID | Int | The ID of the message |
| detailsCorrect | String | 'Y' if the details are correct. If not, submit 'N' and fill in the correct information in the appropriate field(s). |
| correctBirthDate | DateTime | The customer's birth date |

Table 221: SubmitAddressCheckMessage

| Parameter | Type | Description |
|----------------|--------|---|
| messageID | Int | The ID of the message |
| detailsCorrect | String | 'Y' if the details are correct. If not, submit 'N' and fill in the correct information in the appropriate field(s). |
| newAddress1 | String | The first line of the address |
| newAddress2 | String | The second line of the address |
| newAddress3 | String | The third line of the address |
| newTown | String | The town of the address |
| newCounty | String | The county of the address |
| newZipCode | String | The postal code of the address |
| newCountry | String | The country of the address |

Table 222: SubmitContactDetailsCheckMessage

| Parameter | Type | Description |
|--------------------|--------|---|
| messageID | Int | The ID of the message |
| detailsCorrect | String | 'Y' if the details are correct. If not, submit 'N' and fill in the correct information in the appropriate field(s). |
| newHomeTelephone | String | The home telephone number |
| newWorkTelephone | String | The work telephone number |
| newMobileTelephone | String | The mobile telephone number |
| newEmailAddress | String | The e-mail address |

Table 223: SubmitChatNameChangeMessage

| Parameter | Type | Description |
|-------------|--------|----------------------------------|
| messageID | Int | The ID of the message |
| newChatName | String | The customer's desired chat name |

The SubmitLIMBMessage service can be used to send an array of SubmitCardBillingAddressCheckLIMBMessage for each billing card.

Table 224: ArrayOfSubmitCardBillingAddressCheckLIMBMessage

| Parameter | Type | Description |
|----------------|----------|---|
| messageID | Int | The ID of the message |
| detailsCorrect | String | 'Y' if the details are correct. If not, submit 'N' and fill in the correct information in the appropriate field(s). |
| nickName | DateTime | The card nickname |
| newAddress1 | String | The card billing address line one |
| newAddress2 | String | The card billing address line two |
| newAddress3 | String | The card billing address line three |
| newTown | String | The card billing address town |
| newCounty | String | The card billing address county |
| newZipCode | String | The card billing address postal code |
| newCountry | String | The card billing address country |

Output

The following table describes the parameters returned from the submitLIMBMessage service.

Table 225: 1 Instance of SubmitLIMBMessageResp

| Parameter | Type | Description |
|------------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| errorCode | see ... | Service specific error code. See below |
| minorErrorCode | String | Reserved for future use. Currently always null. |
| validationErrors | see ... | An array containing validation errors for the submitted message responses |

Error Codes

The following table describes the error codes returned by submitLIMBMessage.

Table 226: SubmitLIMBMessageErrorEnum

| Value | Condition |
|-------------------|---|
| OK | |
| VALIDATION_ERRORS | There were problems with the submitted information. Check the validationErrors array for specific problems. |
| INVALID_PASSWORD | The submitted password is not valid |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |

The submitLIMBMessage service returns LIMBValidationErrorsEnum as an array. The table below shows the fields in the array of LIMBValidationErrorsEnum.

Table 227: LIMBValidationErrorsEnum

| Value | Condition |
|------------------------------|--|
| INVALID_DOB | The birth date was not valid |
| INVALID_ADDRESS_LINE1 | The address line was not valid |
| INVALID_ADDRESS_LINE2 | The address line was not valid |
| INVALID_ADDRESS_LINE3 | The address line was not valid |
| INVALID_CITY | The city was not valid |
| INVALID_COUNTY_STATE | The county was not valid |
| INVALID_COUNTRY_OF_RESIDENCE | The country was not valid, or a country that is not permitted. |

| Value | Condition |
|--|---|
| INVALID_POSTCODE | The post code was not valid |
| INVALID_HOME_PHONE | The phone number was not valid |
| INVALID_WORK_PHONE | The phone number was not valid |
| INVALID_MOBILE_PHONE | The phone number was not valid |
| INVALID_EMAIL | The e-mail address was not valid |
| INVALID_PASSWORD | The password was incorrect |
| RESERVED_PASSWORD | The desired password is a reserved word and cannot be used |
| INVALID_NEW_PASSWORD | The desired password was blank or not a valid string |
| INVALID_TC_VERSION | The terms and conditions version is not the most recent available |
| INVALID_PRIVACY_VERSION | The privacy policy version is not the most recent available |
| INVALID_CHATNAME | The desired chat name is blank or an invalid string |
| CHATNAME_ALREADY_TAKEN | The desired chat name is already used. |
| INVALID_CARD_BILLING_ADDRESS_LINE_1 | The address line was not valid |
| INVALID_CARD_BILLING_ADDRESS_LINE_2 | The address line was not valid |
| INVALID_CARD_BILLING_ADDRESS_LINE_3 | The address line was not valid |
| INVALID_CARD_BILLING_CITY | The city was not valid |
| INVALID_CARD_BILLING_COUNTY_STATE | The county was not valid |
| INVALID_CARD_BILLING_ZIP_CODE | The post code was not valid |
| INVALID_CARD_BILLING_COUNTRY_OF_RESIDENCE | The country was not valid, or a country that is not permitted. |
| NO_SUCH_PERSONAL_MESSAGE | No message of this type with the submitted ID was found |
| NO_SUCH_TC_PRIVACY_POLICY_MESSAGE | No message of this type with the submitted ID was found |
| NO_SUCH_PASSWORD_CHANGE_MESSAGE | No message of this type with the submitted ID was found |
| NO_SUCH_BIRTH_DATE_CHECK_MESSAGE | No message of this type with the submitted ID was found |
| NO_SUCH_ADDRESS_CHECK_MESSAGE | No message of this type with the submitted ID was found |
| NO_SUCH_CONTACT_DETAILS_CHECK_MESSAGE | No message of this type with the submitted ID was found |
| NO_SUCH_CHATNAME_CHENGE_MESSAGE | No message of this type with the submitted ID was found |
| NO_SUCH_CARD_BILLING_ADDRESS_CHECK_MESSAGE | No message of this type with the submitted ID was found |

| Value | Condition |
|--|---|
| INVALID_PERSONAL_MESSAGE_ACKNOWLEDGMENT | The message acknowledgement was not a valid response. Must be Y or N. |
| INVALID_TC_PRIVACY_POLICY_MESSAGE_ACKNOWLEDGMENT | The message acknowledgement was not a valid response. Must be Y or N. |
| INVALID_BIRTH_DATE_CHECK_MESSAGE | The message acknowledgement was not a valid response. Must be Y or N. |
| INVALID_ADDRESS_CHECK_MESSAGE | The message acknowledgement was not a valid response. Must be Y or N. |
| INVALID_CONTACT_DETAILS_CHECK_MESSAGE | The message acknowledgement was not a valid response. Must be Y or N. |
| INVALID_CARD_BILLING_ADDRESS_CHECK_MESSAGE | The message acknowledgement was not a valid response. Must be Y or N. |

Transfer Funds (global)

The TransferFunds service is for transferring funds between your UK and Australian account wallets. The concept of account wallets has been introduced in release 5.0 of the Betfair API. Instead of a single account holding all of a customer's funds for betting on sports events, there are now two "wallets" for each customer's account: one for betting on the UK exchange server and one for betting on the Australian exchange server.

For information about the introduction of wallets as part of the process of commissioning a dedicated Australian exchange server, see [Betting on Australian Events](#).

Service Details

SOAP Action: transferFunds

Input

The following table describes the parameters used for calling the TransferFunds service.

Table 228: 1 instance of TransferFundsReq

| Parameter | Mandatory? | Type | Description |
|----------------|------------|----------------------------------|--|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| sourceWalletId | Y | Integer | The wallet that you are requesting the funds to be transferred from. There are two possible wallets: 1 = UK Sports Betting wallet 2 = Australian Sports Betting wallet |
| targetWalletId | Y | Integer | The wallet that you are requesting |

| Parameter | Mandatory? | Type | Description |
|-----------|------------|--------|---|
| amount | Y | Double | the funds to be transferred into. There are two possible wallets: 1 = UK Sports Betting wallet 2 = Australian Sports Betting wallet The amount of money you want to transfer. |

Output

The following table describes the parameters returned from the TransferFunds service.

Table 229: 1 Instance of TransferFundsResp

| Parameter | Type | Description |
|---------------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| minorErrorCode | String | This field contains <code>age.verification.required</code> if the user has not completed the age verification process with Betfair. |
| monthlyDepositTotal | Double | The total value of all deposits into the specified wallet so far in the current month. If the transfer request is successful, the value of <code>depositsThisMonth</code> includes the sum that you specified in the transfer request. |
| errorCode | see ... | Service-specific error message. See below. |

Error Codes

The following table describes the error codes returned by TransferFund

Table 230: TransferFundsErrorEnum

| Value | Condition |
|----------------|--|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader <code>errorCode</code> field for additional information |
| INVALID_AMOUNT | The amount to transfer must be greater than 0 |

| Value | Condition |
|---------------------------------|---|
| OVER_BALANCE | The amount you specified to transfer is greater than the amount in the source wallet |
| SOURCE_WALLET_KYC_DEPOSIT_MONTH | During an initial setup period there is a limit on the amount you can transfer into your wallet. If you receive this error message it means that you are attempting to transfer more than the limit into your wallet while the limit is still in force. Contact the Betfair helpdesk. |
| SOURCE_WALLET_KYC_DEPOSIT_TOTAL | During an initial setup period there is a limit on the amount you can transfer into your wallet. If you receive this error message it means that you are attempting to transfer more than the limit into your wallet while the limit is still in force. Contact the Betfair helpdesk |
| SOURCE_WALLET_KYC_WITHDRAWAL | Until Betfair have received and processed the personal information they have requested from you, withdrawals from this wallet are not allowed. |
| SOURCE_WALLET_SUSPENDED | The account associated with the wallet is currently suspended. Please contact the Betfair helpdesk. |
| SOURCE_WALLET_SUSPENDED_KYC | The account associated with the wallet is currently suspended due to Know Your Customer requirements. Please contact the Betfair helpdesk. |
| SOURCE_WALLET_UNKNOWN | The wallet specified does not exist. You must specify 1 for the UK wallet or 2 for the Australian wallet. |
| TARGET_WALLET_KYC_DEPOSIT_MONTH | During an initial setup period there is a limit on the amount you can transfer into your wallet. During this initial setup period, Betfair is legally obliged to request certain personal information from you to validate your identity. When the company has done this, the deposit limit is lifted. If you receive this error message it means that you are attempting to transfer more than the limit into your wallet while the limit is still in force. |
| TARGET_WALLET_KYC_DEPOSIT_TOTAL | During an initial setup period there is a limit on the amount you can transfer into your wallet. If you receive this error message it means that you are attempting to transfer more than the limit into your wallet while the limit is still in force. Contact the Betfair helpdesk |
| TARGET_WALLET_KYC_WITHDRAWAL | Until Betfair have received and processed the personal information they have requested from you, withdrawals from this wallet are not allowed. |
| TARGET_WALLET_SUSPENDED | The account associated with the wallet is currently suspended. Please contact the Betfair helpdesk. |
| TARGET_WALLET_SUSPENDED_KYC | The account associated with the wallet is currently suspended due to Know Your Customer requirements. Please contact the Betfair helpdesk. |

| Value | Condition |
|---------------------------|--|
| TARGET_WALLET_UNKNOWN | The wallet specified does not exist. You must specify 1 for the UK wallet or 2 for the Australian wallet. |
| TRANSFER_FAILED | The transfer would leave the source account over exposed or over the account loss limit, or there is a serious error with the exchange |
| WALLETS_MUST_BE_DIFFERENT | You have specified the same wallet as both the source and the destination for the transfer |

Update Payment Card (global)

The API UpdatePaymentCard service allows you to update the details of a previously registered payment card.

Service Details

SOAP Action: `updatePaymentCard`

Input

The following table describes the parameters used for calling the updatePaymentCard service.

Table 231: 1 instance of UpdatePaymentCardReq

| Parameter | Mandatory? | Type | Description |
|-------------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| nickName | Y | String | The reference name the card was registered under |
| password | Y | String | The Betfair account login password |
| expiryDate | Y | String | The expiry date of the card |
| startDate | N | String | The start date of the card, if applicable for the type of card |
| issueNumber | Y | String | The card's issue number. For all cards except Switch and Solo cards, the value needs to be NULL (this is why the parameter is a string and not an integer). |

| Parameter | Mandatory? | Type | Description |
|------------|------------|---------------------------------------|---|
| address1 | Y | String | The first line of the billing address for the card account |
| address2 | N | String | The second line of the billing address for the card account |
| address3 | N | String | The third line of the billing address for the card account |
| address4 | N | String | The fourth line of the billing address for the card account |
| town | Y | String | The town in which the cardholder resides. |
| county | Y | String | The county in which the cardholder resides. |
| zipCode | Y | String | The postal code of the billing address for the card account |
| country | Y | String | The country of the billing address for the card account |
| cardStatus | Y | PaymentCardStatusEnum | The card's status. It can be either LOCKED or UNLOCKED (see Chapter 61). This parameter is mandatory and is required by Betfair for its internal procedures. If the status submitted in the request is different from the card's actual status, the API returns an UNAUTHORIZED error. To find out the status of the card, use the GetPaymentCard service (see Betfair Simple Data Types6). |

Output

The following table describes the parameters returned from the updatePaymentCard service.

Table 232: 1 Instance of UpdatePaymentCardResp

| Parameter | Type | Description |
|----------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| errorCode | see ... | Service specific error message. See below. |
| minorErrorCode | String | Reserved for future use - currently always nulls |
| nickName | String | An arbitrary name to use for referring to this card |
| billingName | String | The name of the person listed as the account billing name |
| cardType | CardTypeEnum | The type of card (Visa, Mastercard, etc.) |
| expiryDate | String | The expiry date of the card, in the form of MMY |
| startDate | String | The start date of the card, if applicable for the type of card, in the form of MMY |
| address1 | String | The first line of the billing address for the card account |
| address2 | String | The second line of the billing address for the card account |
| address3 | String | The third line of the billing address for the card account |
| address4 | String | The fourth line of the billing address for the card account |
| zipCode | String | The postal code of the billing address for the card account |
| country | String | The country of the billing address for the card account |

Error Codes

The following table describes the error codes returned by UpdatePaymentCard.

Table 233: UpdatePaymentCardErrorEnum

| Value | Condition |
|------------------|---|
| OK | |
| ACCOUNT_INACTIVE | The Betfair account is suspended or locked |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |

| Value | Condition |
|----------------------|---|
| CARD_NOT_FOUND | The referenced card is not registered on the Betfair account |
| INVALID_CARD_ADDRESS | The address did not appear to be a valid address |
| INVALID_CARD_DETAILS | The reference name for the card does not appear to be valid |
| INVALID_COUNTRY_CODE | The country code does not appear to be valid |
| INVALID_EXPIRY_DATE | The expiry date does not appear to be valid (or is too soon) |
| INVALID_ISSUE_NUMBER | The issue number submitted is not valid for the type of card specified. The issue number for that type of card needs to be a two-digit integer. |
| INVALID_PASSWORD | The password is not valid |
| INVALID_START_DATE | The start date does not appear to be valid |
| INVALID_ZIP_CODE | The post code does not appear to be valid |
| UNAUTHORIZED | The Betfair account is not authorised to manage payment cards. (This error is also returned if the user submits a cardStatus parameter that does not agree with the actual cardStatus.) |

View Profile (global)

The API ViewProfile service allows you to retrieve information about the user account, such as the registered address, e-mail address, phone numbers, etc.

Service Details

SOAP Action: `viewProfile`

Input

The following table describes the parameters used for calling the viewProfile service.

Table 234: 1 instance of ViewProfileReq

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |

Output

The following table describes the parameters returned from the viewProfile service.

Table 235: 1 Instance of ViewProfileResp

| Parameter | Type | Description |
|--------------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| minorErrorCode | String | Reserved for future use. Currently always nil. |
| errorCode | | Service specific error code. See below. |
| title | String | The account name title, such as Mr., Ms., Dr., etc. |
| firstName | String | The first name of the account holder |
| surname | String | The surname of the account holder |
| userName | String | The Betfair username for the account |
| forumName | String | The Betfair forum chat name |
| address1 | String | The first line of the account address |
| address2 | String | The second line of the account address |
| address3 | String | The third line of the account address |
| townCity | String | The town or city of the account address |
| countyState | String | The county or state of the account address |
| postCode | String | The postal code of the account address |
| countryOfResidence | String | The country of the account address |
| homeTelephone | String | The home telephone number of the account holder |
| workTelephone | String | The work telephone number of the account holder |
| mobileTelephone | String | The mobile telephone number of the account holder |
| emailAddress | String | The registered e-mail address for the account |
| timeZone | String | The account holder's time zone |
| currency | String | The currency used for the account |

| Parameter | Type | Description |
|---------------------------|----------|--|
| gamcareLimit | Integer | The Gamcare deposit limit for the account. (This parameter corresponds to the depositLimit parameter in the CreateAccount and ModifyProfile services.) |
| gamcareFrequency | | Whether the Gamcare deposit limit is so much per day, per week, per month, or per year. (This parameter corresponds to the depositLimitFrequency parameter in the CreateAccount and ModifyProfile services.) |
| gamcareLossLimit | Integer | The Gamcare loss limit for the account. (This parameter corresponds to the lossLimit parameter in the CreateAccount and ModifyProfile services.) |
| gamcareLossLimitFrequency | | Whether the Gamcare loss limit is so much per day, per week, per month, or per year. (This parameter corresponds to the lossLimitFrequency parameter in the CreateAccount and ModifyProfile services.) |
| gamcareUpdateDate | DateTime | The date of the last change to the account's Gamcare limits. |

Error Codes

The following table describes the error codes returned by viewProfile.

Table 236: ViewProfileErrorEnum

| Value | Condition |
|--------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| UNAUTHORIZED | The current user is not permitted to view the account profile |

View Profile V2(global)

The API ViewProfileV2 service allows you to retrieve information about the user account, such as the registered address, e-mail address, phone numbers, etc.

Service Details

SOAP Action: `viewProfile`

Input

The following table describes the parameters used for calling the `viewProfile` service.

Table 237: 1 instance of ViewProfileReq

| Parameter | Mandatory? | Type | Description |
|----------------|------------|---|--|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| requestVersion | N | ViewProfileV2ReqVersionEnum | You can use <code>requestVersion</code> to specify that the response should contain optional fields corresponding to a particular version. This is to allow new fields in the response without requiring existing clients to handle them. Currently, this is only used by Betfair internal applications. |

The following table describes the fields in `ViewProfileV2ReqVersionEnum`

Table 238: ViewProfileV2ReqVersionEnum

| Value | Condition |
|-------|---|
| V1 | Return an optional field containing the user's national identity. |

Output

The following table describes the parameters returned from the `viewProfile` service.

Table 239: 1 Instance of ViewProfileResp

| Parameter | Type | Description |
|-----------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |

| Parameter | Type | Description |
|--------------------|---------|--|
| minorErrorCode | String | Reserved for future use. Currently always nil. |
| errorCode | | Service specific error code. See below. |
| title | String | The account name title, such as Mr., Ms., Dr., etc. |
| firstName | String | The first name of the account holder |
| surname | String | The surname of the account holder |
| userName | String | The Betfair username for the account |
| forumName | String | The Betfair forum chat name |
| address1 | String | The first line of the account address |
| address2 | String | The second line of the account address |
| address3 | String | The third line of the account address |
| townCity | String | The town or city of the account address |
| countyState | String | The county or state of the account address |
| postCode | String | The postal code of the account address |
| countryOfResidence | String | The country of the account address |
| homeTelephone | String | The home telephone number of the account holder |
| workTelephone | String | The work telephone number of the account holder |
| mobileTelephone | String | The mobile telephone number of the account holder |
| emailAddress | String | The registered e-mail address for the account |
| timeZone | String | The account holder's time zone |
| currency | String | The currency used for the account |
| gamcareLimit | Integer | The Gamcare deposit limit for the account. (This parameter corresponds to the depositLimit parameter in the CreateAccount and ModifyProfile services.) |

| Parameter | Type | Description |
|---------------------------|-----------------------------|--|
| gamcareFrequency | | Whether the Gamcare deposit limit is so much per day, per week, per month, or per year. (This parameter corresponds to the depositLimitFrequency parameter in the CreateAccount and ModifyProfile services.) |
| gamcareLossLimit | Integer | The Gamcare loss limit for the account. (This parameter corresponds to the lossLimit parameter in the CreateAccount and ModifyProfile services.) |
| gamcareLossLimitFrequency | <i>GamcareLimitFreqEnum</i> | Whether the Gamcare loss limit is so much per day, per week, per month, or per year. (This parameter corresponds to the lossLimitFrequency parameter in the CreateAccount and ModifyProfile services.) |
| gamcareUpdateDate | DateTime | The date of the last change to the account's Gamcare limits. |
| tAN | String | The user's TAN |
| referAndEarnCode | String | The user's Refer and Earn Code |
| earthportID | String | The user's Earthport ID |
| KYCStatus | <i>KYCStatusEnum</i> | The status of the Know Your Customer verification for this user |
| nationalIdentifier | String | If requestVersion contains V1, this field is returned containing the national identifier set for the user's account profile. This is currently used only by Betfair to store the national identity number for Indian and South African customer accounts. See: http://en.wikipedia.org/wiki/Permanent_account_number and http://en.wikipedia.org/wiki/National_identification_number#South_Africa |

The following table describes the fields in KYCStatusEnum.

Table 240: KYCStatusEnum

| Value | Condition |
|--------------|---|
| DEFAULT | The user has not been age verified or KYC'd |
| AGE_VERIFIED | The user's age has been verified, but that have not been KYC'd |
| KYC | The user has been fully KYC's and has full access to transfer funds to a from their Aus wallet |
| KYC_NON_AUS | The user has provided some documentation, but the documentation has not been certified by a third-party. The user can transfer funds into their AUS wallet, but they cannot withdraw funds out of their AUS wallet. |

Error Codes

The following table describes the error codes returned by viewProfile.

Table 241: ViewProfileErrorEnum

| Value | Condition |
|--------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| UNAUTHORIZED | The current user is not permitted to view the account profile |

View Refer And Earn(global)

The API ViewReferAndEarn service allows you to retrieve your Refer and Earn code.

Service Details

SOAP Action: viewReferAndEarn

Input

The following table describes the parameters used for calling the ViewReferAndEarn service.

Table 242: 1 instance of ViewReferAndEarnReq

| Parameter | Mandatory? | Type | Description |
|-----------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |

Output

The following table describes the parameters returned from the ViewReferAndEarn service.

Table 243: 1 Instance of ViewReferAndEarnResp

| Parameter | Type | Description |
|------------------|-----------------------------------|---|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| errorCode | see ... | Service specific error code |
| minorErrorCode | String | Additional error messages |
| referAndEarnCode | String | The refer and earn code |

Error Code

The following table describes the error codes returned by ViewReferAndEarn.

Table 244: ViewReferAndEarnErrorEnum

| Value | Condition |
|------------|---|
| OK | |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| NO_RESULTS | No results were returned for the request arguments |

Withdraw To Payment Card (global)

The API WithdrawToPaymentCard service allows you to withdraw funds from your UK wallet using a previously registered payment card. You cannot withdraw funds directly from your Australian wallet. The transferFunds service is provided to enable you to move funds from your Australian wallet into your UK wallet (), and you can then use the WithdrawToPaymentCard service to remove them.

Service Details

SOAP Action: withdrawToPaymentCard

Input

The following table describes the parameters used for calling the withdrawToPaymentCard service.

Table 245: 1 instance of WithdrawToPaymentCardReq

| Parameter | Mandatory? | Type | Description |
|----------------|------------|----------------------------------|---|
| header | Y | APIRequestHeader | The API Request Header that contains the user's session token (see Betfair Simple Data Types). |
| amount | Y | Double | The amount of funds to withdraw |
| cardIdentifier | Y | String | The nickname for the card |

| Parameter | Mandatory? | Type | Description |
|-----------|------------|--------|----------------------|
| password | Y | String | The account password |

Output

The following table describes the parameters returned from the withdrawToPaymentCard service.

Table 246: 1 Instance of WithdrawToPaymentCardResp

| Parameter | Type | Description |
|-----------------|-----------------------------------|--|
| header | APIResponseHeader | The API Response Header that contains the updated session token (see Betfair Simple Data Types). |
| amountWithdrawn | Double | The amount successfully withdrawn to the payment card |
| errorCode | see ... | Service specific error message. See below. |
| maxAmount | Double | The maximum amount you can withdraw to this payment card in a single transaction. |
| minorErrorCode | String | This field contains age.verification.required if the user has not completed the age verification process with Betfair. This field contains PaymentsErrorEnum.UNAUTHORIZED if the user has not completed the identity checking process with Betfair. |

Error Codes

The following table describes the error codes returned by withdrawToPaymentCard. The PaymentsErrorEnum table applies to both deposits and withdrawals so not all enums apply to deposits.

Table 247: PaymentsErrorEnum

| Value | Condition |
|---------------------------|---|
| OK | |
| ACCOUNT_SUSPENDED | The Betfair account is currently suspended |
| API_ERROR | General API error. Check the APIResponseHeader errorCode field for additional information |
| CARD_AMOUNT_OUTSIDE_LIMIT | The deposit amount is lower or higher than the allowed amount for the card |
| CARD_EXPIRED | The card has an expiration date in the past |
| CARD_LOCKED | The card is locked for security reasons |
| CARD_NOT_FOUND | The card is not registered |

| Value | Condition |
|-------------------------------|---|
| CFT_MAX_WITHDRAWAL_LIMIT | The amount available to withdraw has been exceeded |
| DEPOSIT_DECLINED | The deposit was declined |
| DEPOSIT_LIMIT_EXCEEDED | The deposit exceeded the GamCare deposit limit amount |
| DUPLICATE_WITHDRAWAL | This error is returned if two withdrawals are made to the same card within 10 seconds of each other. (The API refuses the second request, because of the possibility that it has resulted from a double-click by the user of the client application.) |
| EXCEEDS_BALANCE | The deposit exceeds the amount available on the card |
| INVALID_AMOUNT | The amount entered is not a valid figure |
| INVALID_CARD_CV2 | The CV2 digits are not valid |
| INVALID_CARD_DETAILS | The card was not recognised |
| INVALID_EXPIRY_DATE | The expiration date was not a valid date |
| INVALID_MASTERCARD | Withdrawal to a non-UK Mastercard not allowed until the card is validated by Betfair |
| INVALID_PASSWORD | The account password is invalid |
| NEGATIVE_NET_DEPOSITS | The amount to withdraw is more than the amount deposited from a credit card |
| NON_STERLING_TO_UK_MASTERCARD | The currency of the account is not in Sterling and the card is a UK sterling Mastercard. |
| NON_ZERO_NON_NEG_NET_DEPOSITS | The account has funds deposited from other cards. You must withdraw the amount deposited from the other card(s) before you can withdraw to this card. |
| UNAUTHORIZED | The account is not authorised to manage payment cards |
| VISA_WITHDRAWAL_NOT_POSSIBLE | The VISA card policy does not allow withdrawal to this card. |

Appendix

Additional Information

Time zones

All times are returned in GMT. They can be converted to your local timezone or into the local market timezone using the timezone returned by the market field. The API will return the code listed in the Abbreviation column if available.

The following table lists the timezone codes returned by the API along with their meaning.

Table 248: Valid Timezones

| 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 | GMT | Greenwich Mean Time |
| Hongkong | HK | |
| Jamaica | KMT | |
| Japan | JPT | |
| NZ | NZT | New Zealand |
| US/Alaska | AKST | |

| Location | Abbreviation | Notes |
|-------------|--------------|-------|
| US/Arizona | AST | |
| US/Central | CST | |
| US/Eastern | EST | |
| US/Hawaii | HST | |
| US/Mountain | MST | |
| US/Pacific | PST | |

Currency

All currency amounts shown will be in the account currency. This can be overridden in the case of RefreshMarket. Specify a different currency in the header to have amounts returned in a different currency.

You can determine the list of supported currencies, the ISO3 Currency Code and the minimum bet size for the supported currencies from the getAllCurrenciesV2 service.

Line and Range Markets

From release 5.0, line and range markets are supported by the API.

Timestamps

We recommend using a local Network Time Protocol server to synchronize your computer's clock with Betfair timestamps. We recommend that you synchronise with the one that your ISP provides or go to <http://www.pool.ntp.org/en/> to find the nearest NTP server to you.

Horse Racing Non-Runners

In Horse Racing terms, any horse declared for a race which does not subsequently come under starter's orders is deemed to be a non-runner. Prior to all betting on a race, each horse is given a 'reduction factor', based on Betfair's estimate of each horse's chance of winning with reference to the racing press - this number can be retrieved using the API via the MarketInfo field in the response to the MarketRefresh call. In general the higher the number the more fancied the horse. Reductions will be made to both win and place markets but applied differently, and horses will have a different reduction factor for each.

On win markets the reduction factor is applied to the total odds. Thus, if a reduction factor of 12% is applied to odds of 5, the resulting reduced odds would be $5 - (12\% \times 5) = 4.4$. In this case, it is mathematically provable that the reduction factors must sum to 100%, if they are to be fair.

However, for place markets, the reduction factors are applied to the odds minus the stakes (i.e. the odds-1) so that if a 40% RF is applied to a bet at odds of 3, it gets reduced to $3 - ((3-1) \times 40\%) = 2.2$. In this case, the sum of the reduction factors do not sum to 100%, or any specific value.

The calculation to get these reduction factors is based on predicted odds (from a reputable source such as the Racing Post) and uses an iterative process to calculate the 'fair' reduction factor based on the effect of removing each of the runners in turn. In other words, it removes a runner and calculates what the reduction factor would have to be in order to return a 'fair' market back to 100%, after the removal. For this purpose a 'fair' market is assumed to be one where the market reflects the reduction factors precisely.

The precise calculation used is proprietary.

Betting In-Running

When a market is In-Running, or In-Play, it signifies that the event the market refers to, or a closely related event, has begun. The In-Running status can be determined by checking the delay field in the response to the GetMarketPrices

call. When the market is in-running, the delay non-zero. Other services also return the bet delay and can be used to determine the in-play status of a market.

When a market is In-Running, betting on this market operates in an identical fashion to normal, with one difference. When you place your bet there is a deliberate time delay before the bet is processed. This is to protect users who have left orders on the exchange and wish to cancel or edit them when something dramatic happens during the event. There is no time delay for cancelling bets.

The time delay for editing a bet depends on the type of edit you want to perform. The following table shows the possible ways to edit a bet and the amount of delay for each:

Table 249: Editing Bets and Resulting Time Delay

| If you... | The resulting delay is: |
|--------------------|--|
| Reduce the stake | None. The edit happens immediately. |
| Increase the stake | The delay is applied. |
| Change the price | The original bet is cancelled immediately. A new bet at the new price is placed after the delay. |

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 format of the Locale string is:

```
<language code>[_<REGION CODE>[_<variant>]]
```

The language and region code are used to determine the appropriate locale. The variant is reserved for future use.

The language code is based on the ISO 639-1 standard which defines two-letter codes, such as "en" and "fr". If a two-letter code is not available, then ISO 639-2 three-letter identifiers are accepted as well. For more details, see <http://lcweb.loc.gov/standards/iso639-2/englangn.html>.

The region code is defined by ISO 3166-1 (see <http://www.iso.org/iso/en/prods-services/iso3166ma/04background-on-iso-3166/what-is-iso3166.html>). The region code is two-letter, in all capital letters, and appended after an underscore after the language code.

Table 250: Locale String Examples

| Locale String | Locale |
|---------------|---|
| en | English language |
| en_GB | English language with United Kingdom region |
| haw | Hawaiian language |
| en_Variant | NOT VALID |

Exponential Numbers

The API uses the type Double to represent various values, such as price and traded volume. The XML standard definition of the type Double specifies that values of type Double can be represented by exponential notation (for example, 1.0E7). When numbers reach around 10 million, the API returns the values in exponential notation.

The common SOAP libraries, such as Java Axis and .Net automatically parse values of type Double regardless of how they are represented in the XML response. However, if you are using any of the compressed services, the values returned in the compressed response are seen as Strings.

You must make sure your application can correctly handle values returned in the compressed response string that are represented using exponential notation.

For more information on the XML type Double, see <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/#double>.

Betfair Simple Data Types

In order to provide comprehensive validation on certain fields we have provided simple enumerated data types for some of the input and output parameters. This section details the types we have used.

AccountStatementEnum

The following table describes the AccountStatementEnum simple data type.

Table 251: AccountStatementEnum

| Value | Description |
|-----------------------|---|
| OK | |
| RESULT_ERR | Internal Error |
| RESULT_FIX | Result has been updated after an initial state. I.e. your account history has been changed to reflect this. |
| RESULT_LOST | Loss |
| RESULT_NOT_APPLICABLE | The record has no applicable result (e.g. commission record) |
| RESULT_WON | Won |
| COMMISSION_REVERSAL | Betfair have restored funds to your account that it previously received from you in commission. |

AccountStatementIncludeEnum

The following table describes the AccountStatementIncludeEnum simple data type.

Table 252: AccountStatementIncludeEnum

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

AccountStatusEnum

the following table describes the AccountStatusEnum simple data type.

Table 253: AccountStatusEnum

| Value | Description |
|-------|---|
| A | Account active |
| C | Account closed |
| D | Account suspended - duplicate cards |
| L | Account is locked |
| P | Account is pending authorization |
| S | Account is suspended |
| T | Account requires Telbet Terms and Conditions acceptance |
| X | Account suspended - failed security questions |
| Z | Account suspended - KYC check failure |

AccountTypeEnum

The following table describes the AccountTypeEnum simple data type.

Table 254: AccountTypeEnum

| Value | Description |
|--------------|--|
| STANDARD | A standard Betfair account |
| MARGIN | A margin account |
| TRADING | A trading account under a Master/Sub account |
| AGENT_CLIENT | An account created by an agent on behalf of a client |

BetCategoryTypeEnum

The following table describes the BetCategoryTypeEnum simple data type.

Table 255: BetCategoryTypeEnum

| Value | Description |
|-------|--|
| NONE | Equivalent to E. This may change in a future release. |
| E | Normal exchange bet. |
| M | Market on Close bet. The bet remains unmatched until the market is reconciled and a starting price is determined. If no starting price is available for the selection, the bet lapses. |
| L | Limit on Close bet. The bet remains unmatched until the market is reconciled and a starting price is determined. If the starting price is better than the price specified, then the bet is matched. If no starting price is available for the selection, the bet lapses. |

BetPersistenceTypeEnum

The following table describes the BetPersistenceTypeEnum simple data type.

Table 256: BetPersistenceTypeEnum

| Value | Description |
|-------|--|
| NONE | Normal exchange or SP bet. Unmatched exchange bets are lapsed when the market turns in-play. |
| IP | In Play persistence. Unmatched bets (or the unmatched portion of a bet) remain in the market when it turns in-play. The bet retains its place in the bet queue and retains the same betId. |
| SP | MoC Starting Price. The Unmatched (or unmatched portion) bet is converted to a Market on Close Starting Price bet and matched at the reconciled starting price. |

BetsOrderByEnum

The following table describes the BetsOrderByEnum simple data type.

Table 257: BetsOrderByEnum

| Value | Description |
|----------------|-------------------------|
| BET_ID | Order by Bet ID |
| CANCELLED_DATE | Order by Cancelled Date |
| MARKET_NAME | Order by Market Name |
| MATCHED_DATE | Order by Matched Date |
| NONE | Default order |
| PLACED_DATE | Order by Placed Date |

BetStatusEnum

The following table describes the BetStatusEnum simple data type.

Table 258: BetStatusEnum

| Value | Description |
|-------|-----------------------|
| C | Cancelled |
| L | Lapsed |
| M | Matched |
| MU | Matched and Unmatched |
| S | Settled |
| U | Unmatched |
| V | Voided |

**Note:**

Not all values are valid in all situations. Please check the documentation for the individual services.

BetTypeEnum

The following table describes the BetTypeEnum simple data type.

Table 259: BetTypeEnum

| Value | Description |
|-------|-------------|
| B | Back |
| L | Lay |

BillingPeriodEnum

The following table describes the BillingPeriodEnum simple data type.

Table 260: BillingPeriodEnum

| Value | Description |
|-----------|--------------------------|
| WEEKLY | Weekly billing period |
| MONTHLY | Monthly billing period |
| QUARTERLY | Quarterly billing period |
| ANNUALLY | Annually billing period |

CardTypeEnum

The following table describes the CardTypeEnum simple data type.

Table 261: CardTypeEnum

| Value | Description |
|-------------------|--|
| VISA | Visa credit card |
| MASTERCARD | Mastercard credit card |
| VISADELTA | Visa debit card |
| SWITCH | Switch debit card |
| SOLO | Solo card |
| ELECTRON | Electron card |
| LASER | Laser debit card |
| MAESTRO | Maestro debit card |
| INVALID_CARD_TYPE | The card described in your service request is not one of the valid card types listed in this table |

GamcareLimitFreqEnum

The following table describes the GamCareLimitFreqEnum simple data type.

Table 262: GamcareLimitFreqEnum

| Value | Description |
|---------|--------------------------------------|
| DAILY | Deposit amount limit applies daily |
| WEEKLY | Deposit amount limit applies weekly |
| MONTHLY | Deposit amount limit applies monthly |
| YEARLY | Deposit amount limit applies yearly |

GenderEnum

The following table describes the GenderEnum simple data type.

Table 263: GenderEnum

| Value | Description |
|-------|-------------|
| F | Female |
| M | Male |

MarketStatusEnum

The following table describes the MarketStatusEnum simple data type.

Table 264: MarketStatusEnum

| Value | Description |
|-----------|---|
| ACTIVE | Market is open and available for betting. |
| CLOSED | Market is finalised, bets to be settled. |
| INACTIVE | Market is not yet available for betting. |
| SUSPENDED | Market is temporarily closed for betting, possibly due to pending action such as a goal scored during an in-play match or removing runners from a race. |

MarketTypeEnum

The following table describes the MarketType simple data type.

Table 265: MarketTypeEnum

| Value | Description |
|----------------|--|
| A | Asian Handicap |
| L | Line |
| O | Odds |
| R | Range |
| NOT_APPLICABLE | The market does not have an applicable market type |

MarketTypeVariantEnum

The following table describes the MarketTypeVariant simple data type.

Table 266: MarketTypeVariantEnum

| Value | Description |
|-------|-------------------|
| D | Default |
| ASL | Asian Single Line |
| ADL | Asian Double Line |

PaymentCardStatusEnum

The following table describes the PaymentCardStatusEnum simple data type.

Table 267: PaymentCardStatusEnum

| Value | Description |
|----------|---|
| LOCKED | The card has been locked and cannot be changed or used to deposit or withdraw funds |
| UNLOCKED | The card is unlocked |

RegionEnum

The following table describes the RegionEnum simple data type.

Table 268: RegionEnum

| Value | Description |
|---------|-----------------------|
| AUZ_NZL | Australia/New Zealand |
| GBR | Great Britain |
| IRL | Ireland |
| NA | North America |
| NORD | Nordic |
| ZAF | South Africa |

SecurityQuestion1Enum

The following table describes the SecurityQuestion1Enum simple data type.

Table 269: SecurityQuestion1Enum

| Value | Description |
|-------|------------------------|
| SQ1A | Favourite Sports Team? |
| SQ1B | Pet's name? |
| SQ1C | Favourite Movie? |
| SQ1D | Favourite Food? |

SecurityQuestion2Enum

The following table describes the SecurityQuestion2Enum simple data type.

Table 270: SecurityQuestion2Enum

| Value | Description |
|-------|---------------------|
| SQ2A | Anniversary? |
| SQ2B | Partner's Birthday? |
| SQ2C | Mother's Birthday? |
| SQ2S | Father's Birthday? |

ServiceEnum

The following table describes the ServiceEnum simple data type.

Table 271: ServiceEnum

| Value | Description |
|-------------------------------|--|
| ADD_PAYMENT_CARD | Access to addPaymentCard service |
| CANCEL_BETS | Access to cancelBets service |
| CREATE_ACCOUNT | Access to createAccount service |
| CONVERT_CURRENCY | Access to convertCurrency service |
| DELETE_PAYMENT_CARD | Access to deletePaymentCard service |
| DEPOSIT_FROM_PAYMENT_CARD | Access to depositFromPaymentCard service |
| DO_KEEP_ALIVE | Access to keepAlive service |
| EDIT_BETS | Access to updateBets service |
| FORGOT_PASSWORD | Access to forgotPassword service |
| GET_ACCOUNT_STATEMENT | Access to getAccountStatement service |
| GET_BET | Access to getBet service |
| GET_CURRENT_BETS | Access to getCurrentBets/getMUBets services |
| GET_CURRENCIES | Access to getAllCurrencies service |
| GET_MARKET_TRADED_VOLUME | Access to getMarketTradedVolume service |
| GET_PAYMENT_CARD | Access to getPaymentCard service |
| LOAD_BET_HISTORY | Access to getBetHistory service |
| LOAD_DETAILED_AVAIL_MKT_DEPTH | Access to getDetailAvailableMktDepth service |
| LOAD_EVENT_TYPES | Access to getAllEventTypes service |
| LOAD_EVENTS | Access to getEvents service |
| LOAD_MARKET | Access to getMarkets service |
| LOAD_MARKET_PRICES | Access to getMarketPrices service |
| LOAD_MARKET_PRICES_COMPRESSED | Access to getMarketPricesCompressed service |
| LOAD_MARKET_PROFIT_LOSS | Access to getMarketProfitAndLoss service |

| Value | Description |
|----------------------------|--|
| LOAD_SERVICE_ANNOUNCEMENTS | Reserved for future release - currently null |
| LOAD_SUBSCRIPTION_INFO | Access to getSubscriptionInfo service |
| LOGIN | Access to login service |
| LOGOUT | Access to logout service |
| MODIFY_PASSWORD | Access to modifyPassword service |
| MODIFY_PROFILE | Access to modifyProfile service |
| PLACE_BETS | Access to placeBets service |
| RETRIEVE_LIMB_MESSAGE | Access to retrieveLIMBMessage service |
| SUBMIT_LIMB_MESSAGE | Access to submitLIMBMessage service |
| UPDATE_PAYMENT_CARD | Access to updatePaymentCard service |
| VIEW_PROFILE | Access to viewProfile services |
| WITHDRAW_TO_PAYMENT_CARD | Access to withdrawToPaymentCard service |

SortOrderEnum

The following table describes the SortOrderEnum simple data type.

Table 272: SortOrderEnum

| Value | Description |
|-------|-----------------|
| ASC | Sort ascending |
| DESC | Sort descending |

SubscriptionStatusEnum

The following table describes the SubscriptionStatusEnum simple data type.

Table 273: SubscriptionStatusEnum

| Value | Description |
|-----------|------------------------|
| ACTIVE | Subscription Active |
| INACTIVE | Subscription Inactive |
| SUSPENDED | Subscription Suspended |

TitleEnum

The following table describes the TitleEnum simple data type.

Table 274: TitleEnum

| Value | Description |
|-------|-------------|
| Dr | Dr. |
| Miss | Miss. |
| Mr | Mr. |

| Value | Description |
|-------|-------------|
| Mrs | Mrs. |
| Ms | Ms. |

ValidationErrorsEnum

The following table describes the ValidationErrorsEnum simple data type.

Table 275: ValidationErrorsEnum

| Value | Condition |
|----------------------------------|--|
| DUPLICATE_USERNAME | The username already exists |
| FUNDS_TRANSFER_CANCEL | N/A |
| FUNDS_TRANSFER_CURRENCY_MISMATCH | N/A |
| INCOMPLETE_DETAILS | Not enough information provided |
| INSUFFICIENT_FUNDS | N/A |
| INVALID_ACCOUNT_TYPE | The account to create was not recognized |
| INVALID_ADDRESS_LINE1 | The address line was not a valid string |
| INVALID_ADDRESS_LINE2 | The address line was not a valid string |
| INVALID_ADDRESS_LINE3 | The address line was not a valid string |
| INVALID_ANSWER1 | The answer was not a valid string |
| INVALID_ANSWER2 | The answer was not a valid string |
| INVALID_BROWSER | The browser ID is not valid |
| INVALID_CITY | The City was not recognized |
| INVALID_COUNTRY_OF_RESIDENCE | The Country was not recognized or was a country that is not permitted to create an account |
| INVALID_COUNTY_STATE | The county/state string was not recognized |
| INVALID_CURRENCY | The currency for the account was not recognized |
| INVALID_DEPOSIT_LIMIT | The deposit limit was not a valid number |
| INVALID_DEPOSIT_LIMIT_FREQUENCY | The Gamcare frequency was not recognized |
| INVALID_DETAILS | The account details were not recognized |
| INVALID_DOB | The date of birth was not understood |
| INVALID_EMAIL | The e-mail string was not recognized as a valid e-mail address |
| INVALID_FIRSTNAME | The firstname field was blank or not a string |
| INVALID_GENDER | The gender was not within a valid range |
| INVALID_HOME_PHONE | The phone number was not recognized as a valid number |
| INVALID_IP_ADDRESS | The IP address was not recognized as valid |
| INVALID_LANGUAGE | The language was not recognized as a valid string |
| INVALID_LOCALE | The locale was not recognized |

| Value | Condition |
|------------------------------|--|
| INVALID_LOSS_LIMIT | The loss limit specified was not recognized as a valid integer |
| INVALID_LOSS_LIMIT_FREQUENCY | The Gamcare frequency was not recognized |
| INVALID_MASTER_ID | The master ID was not recognized |
| INVALID_MOBILE_PHONE | The phone number was not recognized as a valid number |
| INVALID_PARTNERID | The partner ID was not recognized |
| INVALID_PASSWORD | The password was not in the correct format. Please see the Betfair website for password rules. |
| INVALID_POSTCODE | The post code was not recognized as valid |
| INVALID_PRIVACY_VERSION | The privacy policy version number was not recognized |
| INVALID_PRODUCT_ID | The ID was blank or not recognized |
| INVALID_REFERRER_CODE | The Refer and Earn code was not recognized |
| INVALID_REGION | The region was not recognized |
| INVALID_SECURITY_QUESTION1 | The security question was not recognized as a valid string |
| INVALID_SECURITY_QUESTION2 | The security question was not recognized as a valid string |
| INVALID_SUBPARTNERID | The ID was blank or not recognized |
| INVALID_SUPERPARTNERID | The ID was blank or not recognized |
| INVALID_SURNAME | The account surname was blank or not a valid string |
| INVALID_TC_VERSION | The terms and conditions version number was not recognized |
| INVALID_TIMEZONE | The time zone was blank or not recognized |
| INVALID_TITLE | The title was blank or not recognized |
| INVALID_USERNAME | The username was blank or not recognized |
| INVALID_WORK_PHONE | The phone number was not recognized as a valid number |
| RESERVED_PASSWORD | The password matches a reserved word and is not allowed |