

Betfair API 5.2.2

SOAP API Reference Guide

Document Version: 1.0

Copyright and Terms

This guide and the Betfair API are © The Sporting Exchange Limited 2006, all rights reserved

TERMS AND CONDITIONS

1. DEFINITIONS AND INTERPRETATION

1.1 **"Agreement"** means the Term Sheet, these Terms and Conditions and any associated schedules;

"Betfair API" means the data and functionality (commonly referred to as the application programming interface or API) contained on the Betfair betting exchange platform together with access to the relevant Web Services Definition Language (WSDL) file where required;

"Betfair Terms and Conditions " means the terms and conditions, rules and regulations and privacy policy governing the use of the www.betfair.com web site as available for inspection at that site;

"Business Day" means any day (excluding Saturdays and Sundays) on which banks generally are open in the City of London for the transaction of normal banking business;

"Code" means encrypted data that enables Betfair to identify circumstances in which a particular software application is in use;

"Commencement Date" means the date on which this Agreement is agreed to by both parties;

"Commercialise" means sell, rent, lease, license or publish all or any part of the Betfair API for profit or otherwise use in a commercial or business context that does not equate to personal use or enable others to do so;

"Confidential Information" means all information which is not publicly known and that is disclosed (by whatever means, directly or indirectly) by one party to the other, whether before or after the date of this Agreement including any information relating to the IPR, products, operations, processes, plans, intentions, product information, each party's customer data the terms of this Agreement, market opportunities or business affairs of the disclosing party or any of its sub-contractors, suppliers, customers, clients or other contacts;

"Content" means all Confidential Information, data and/or functionality accessible from the Betfair API;

"Force Majeure" means any event outside the reasonable control of a party affecting its ability to perform any of its obligations (other than payment) under this Agreement;

"Good Industry Practice" means the exercise of that degree of skill, diligence, prudence and foresight which would reasonably and ordinarily be expected from a skilled and experienced contractor acting in good faith;

"Group" means, in relation to a company, any entity directly or indirectly controlling, controlled by or under common control of a party;

"IPR" means any and all patents, trade marks, service marks, rights in designs (including semi-conductor topography design rights and circuit layout rights), get-up, trade, business or domain names, goodwill associated with the foregoing, e-mail address names, copyright including rights in computer software (in both source and object code) and rights in databases (in each case whether registered or not and any applications to register and rights to apply for registration of any of the foregoing), rights in inventions and web-formatting scripts (including HTML and XML scripts), know-how, trade secrets and other intellectual property rights which may now or in the future subsist in any part of the world including all rights of reversion and the right to sue for and recover damages for past infringements;

"Read Only Access" means access to the Betfair API with all the functionality set out in Schedule 1;

"Transactional Access" means access to the Betfair API with all the functionality set out in Schedule 2;

"UserID" means the confidential security keys specific to the User issued by Betfair to enable access to the Betfair API; and

1.2 Headings to clauses are inserted for convenience only and shall not affect the interpretation or construction of this Agreement.

1.3 Words importing the singular shall include the plural and vice versa. Words importing a gender include every gender and references to persons include an individual, company, corporation, firm or partnership.

1.4 The words and phrases "other", "including" and "in particular" shall not limit the generality of any preceding words or be construed as being limited to the same class as any preceding words where a wider construction is possible.

1.5 Any obligation to do or not to do something shall include an obligation to procure that it be done or not done.

1.6 References to a party include that party's successors and permitted assignees.

1.7 In this Agreement, "control" shall have the meaning given to it in section 840 of the Income and Corporation Taxes Act 1988.

1.8 This Agreement may be executed in any number of counterparts which together shall constitute one agreement. Each party may enter into this Agreement by executing a counterpart and this Agreement shall not take effect until it has been executed by both parties.

1.9 Delivery of an executed counterpart of a signature page by facsimile transmission shall take effect as delivery of an executed counterpart of this Agreement provided that, if such method is adopted, each party shall provide the other with the original of such page as soon as reasonably practicable thereafter.

THE PARTIES AGREE AS FOLLOWS:

2. LICENCE

2.1 In consideration of the payment by the User of the Access Fee in accordance with clause 4 below and subject to the terms and conditions of this Agreement, Betfair hereby grants to the User a non-exclusive, non-transferable, terminable licence to access the Betfair API solely for its own internal business purposes and for the purposes of developing software applications in each case in accordance with such other limitations and restrictions as set out in this Agreement.

2.2 Betfair agrees to enable the User's User ID with access to the Betfair API during the term of this Agreement. The User ID shall remain the property of Betfair and may be disabled from access to the Betfair API upon termination of this Agreement or upon occurrence of any of the events described in clause 2.3 below.

2.3 It is a condition of this Agreement that the User shall not do any of the following:

- (a) assign, transfer, sub-license or disclose the User ID to a third party;
- (b) assign, transfer, sub-license, disclose or otherwise provide any of the Content to a third party;
- (c) collect personally identifiable information of any other user of the Betfair API;
- (d) Commercialise the Content or access to the Betfair Exchange Platform;

- (e) use the API in a manner which might enable a third party to interact with the web site www.betfair.com or any other web site owned or operated by Betfair;
- (f) use the Content to provide market information to a third party for business or commercial use;
- (g) display data from the Betfair API via any electronically accessible medium without the express written consent of Betfair;
- (h) create freeware, shareware or commercial software applications for use in connection with the API without the express written consent of Betfair;
- (i) utilise software applications made available by third parties and intended for use specifically in relation to betting exchange functionality and/or the API other than with the express written consent of Betfair or of a software developer who holds a Developer's API licence and who is in compliance with the terms of such licence including in particular the requirement to use a Code in software applications;
- (j) use the Betfair API in a way which proves or is likely to prove detrimental to Betfair and/or the performance of the web site www.betfair.com.
- (k) copy, reproduce, modify or use the API in any bureau, timeshare, or outsourcing arrangement or reproduce or on sell the whole or any part of the API whether aggregated with other data or otherwise.

2.4 By entering into this Agreement the User hereby agrees to the Betfair Terms and Conditions.

3. WARRANTIES OF THE USER

3.1 The User, warrants and undertakes that:

- (a) it has full capacity and authority and all necessary licences, permits, IPR rights and consents to enter into this Agreement and any other documents executed by it that may be associated with this Agreement;
- (b) this Agreement constitutes valid, binding and enforceable obligations of the User in accordance with its terms;
- (c) it shall at all times conduct itself with all due skill, care and diligence, including Good Industry Practice, and in accordance with its own established procedures and all applicable laws, enactments, orders, regulations and other similar instruments;
- (d) it shall comply with Betfair's security guidelines and requirements as may be issued by Betfair from time to time whether in writing or otherwise.

4. ACCESS FEE

4.1 The User agrees to pay to Betfair an access fee in the sum and at the frequency stated in the Term Sheet (the relevant "Access Fee"). Unless otherwise specified prices shall be exclusive of VAT.

4.2 Nothing in this Agreement shall require Betfair to make any payment to the User with respect to this Agreement.

5. INTELLECTUAL PROPERTY RIGHTS

5.1 All IPR in the Betfair API shall belong to Betfair. All IPR in any third party materials shall belong to the third party owner thereof.

5.2 Nothing in this agreement purports to grant a license, provide any warranty or offer any indemnity in respect of any data that is not owned by Betfair. In the event that the User does require access to any such data, it agrees that it shall enable Betfair an opportunity to secure rights to the same and (if it becomes necessary to do so) the User will cover the costs of securing a licence to the same from the relevant third party data owner or either party may terminate this agreement immediately.

6. INDEMNITY

6.1 The User shall at its own expense indemnify Betfair against any claim against Betfair alleging an infringement by Betfair of the IPR of any third party arising through the User's use of the Betfair API and pay any final judgement entered against Betfair in respect thereof except if and to the extent that any such claim arises from any breach by Betfair of its obligations under this Agreement.

6.2 Either party shall immediately notify the other party if any claim or demand is made or action brought against it for any infringement or alleged infringement of any IPR which may affect the supply or use of the Betfair API.

7. DATA PROTECTION AND SECURITY

7.1 The User will duly observe all its obligations under the Data Protection Act 1998 and any amendments thereto which arise in connection with this Agreement. In particular the User shall ensure that it has adequate technical (and organisational) security procedures in place to prevent the unauthorised or unlawful disclosure of personal data.

7.2 The User acknowledges that the security of Betfair's data and its systems is fundamental to the business of Betfair and if the User becomes aware of a breach or potential breach of security relating to the Betfair API, it shall immediately notify Betfair of such breach or potential breach and use its best endeavours to ensure that any potential breach does not become an actual breach and/or remedy any actual breach and its consequences.

8. CONFIDENTIALITY AND ANNOUNCEMENTS

8.1 During the term of this Agreement and after termination or expiration of this Agreement, the parties shall not use any Confidential Information for any purpose other than in pursuance of its rights and obligations under this Agreement nor disclose any Confidential Information to any person except with the prior written consent of the other party and shall use the same standard of security to prevent the use or disclosure of the Confidential Information as it does for its own Confidential Information.

8.2 The parties may disclose any Confidential Information to their directors, other officers, employees, advisers and sub-contractors to the extent that such disclosure is reasonably necessary and in accordance with the requirements set out in clause 8.1.

8.3 On termination the parties shall (on request) deliver up to the other party or destroy all copies of Confidential Information in its possession, and (if so requested) shall use all reasonable endeavours to destroy all copies of Confidential Information stored electronically.

8.4 The parties shall together determine the content of any communications concerning the relationship between the parties. Such communications shall be issued at a time and in a manner agreed by the parties (acting reasonably).

9. LIMITATION OF LIABILITY AND EXCLUSIONS

9.1 Save as provided by statute and to the fullest extent permitted by law, the following provisions set out the entire liability of Betfair (including any liability for the acts and omissions of its employees, agents and sub-contractors) to the User whether in contract, tort, statute, equity or otherwise:

(a) The User acknowledges and agrees that (except as expressly provided in this Agreement) the Betfair API is provided "AS IS" without warranties of any kind (whether express or implied);

(b) All conditions, warranties, terms and undertakings (whether express or implied, statutory or otherwise) relating to the delivery, performance, quality, uninterrupted use, fitness for purpose, occurrence or reliability of the Betfair API are hereby excluded to the fullest extent permitted by law;

(c)	The entire liability of Betfair in respect of any breach or default shall be limited to £1,000; and				part of its assets, or if Sub-Licensee make an assignment for the benefit of, or a composition or arrangement with, its creditors;
(d)	Betfair shall not be liable to the User for loss of profit (whether direct or indirect), loss of contracts or goodwill, lost advertising, loss of data or any type of special, indirect, consequential or economic loss (including loss or damage suffered by the User as a result of an action brought by a third party) even if such loss was reasonably foreseeable or Betfair had been advised of the possibility of the User incurring such loss.			(d)	the Breaching Party failing to make a payment by such due date as may be specified in this Agreement provided always that in relation to any indebtedness of the Breaching Party, the failure to pay when due which shall be deemed a remediable material breach to be determined in accordance with clause 12.2(b) above;
9.2	No exclusion or limitation set out in this Agreement shall apply in the case of:				(e) there is a change in control of the User without the written consent of Betfair;
(a)	fraud or fraudulent concealment;				(f) the User is in breach of any of the restrictions set out in clause 2.3
(b)	death or personal injury resulting from the negligence of either party or any of its employees, agents or sub-contractors; and/or	12.3			For the purposes of clause 12.2 a " material breach " means a breach which is serious in the widest sense of having a serious effect on the benefit which the Initiating Party would otherwise derive from a substantial portion of this Agreement over the entire remaining period of this Agreement or a reasonable portion thereof and a breach is remediable if the Breaching Party can comply with the obligation within the 10 Business days period in sub-clause 12.2(b).
(c)	any breach of the obligations implied by (as appropriate) section 12 of the Sale of Goods Act 1979, section 2 of the Supply of Goods and Services Act 1982 or section 8 of the Supply of Goods (Implied Terms) Act 1973.	12.4			Betfair may terminate this Agreement on one month's notice in writing at any time for any reason provided always that it returns any Access Fee payments already made by the User that relate to any period that follows such termination date.
9.3	The time limit within which the User must institute suit against Betfair to recover on any claim shall be 2 years from the date the User should reasonably have become aware or becomes aware of the relevant breach that would form the subject of the claim.				
9.4	This clause 9 shall survive the termination of this Agreement for whatever reason.	13.			CONSEQUENCES OF TERMINATION
10.	FORCE MAJEURE	13.1			Termination of this Agreement shall be without prejudice to any rights or obligations which shall have accrued prior to termination.
10.1	If either party is affected by Force Majeure it shall notify the other party in writing of the matters constituting the Force Majeure and shall keep that party informed of their continuance and of any relevant change of circumstances whilst such Force Majeure continues. Neither party shall have any liability to the other in respect of an event of Force Majeure provided it complies with clause 10.2.	13.2			On termination of this Agreement all licences granted by Betfair to the User pursuant to this Agreement shall immediately terminate and the User ID shall be disabled for use in connection with the Betfair API.
10.2	The party affected by Force Majeure shall take all reasonable steps available to it to minimise the effects of Force Majeure on the performance of its obligations under this Agreement.	13.3			Within 10 days of the termination of this Agreement the User shall at Betfair's sole option return or destroy all copies of the Content in its possession or control and a duly authorised officer of the User shall certify in writing to Betfair that the User has complied with this obligation.
11.	TERM	13.4			The expiry or termination of this Agreement for whatever reason shall not affect:
	This Agreement shall commence on the Commencement Date and, unless terminated earlier in accordance with clause 12, shall continue until the User ceases to pay the Access Fee specified in clause 4.1 at which time the User will (provided it gives notice of an intention to renew) have 24 hours to secure continued access by payment of the next instalment of the Access Fee or the Agreement will terminate.	(a)			either party's accrued rights and obligations at the date of expiry or termination;
		(b)			the coming into force or the continuance in force of any provision of this Agreement which expressly or by implication is intended to come into or continue in force on or after such expiry or termination.
12.	TERMINATION	14.			ASSIGNMENT AND SUB-CONTRACTING
12.1	A party (the " Initiating Party ") may terminate this Agreement with immediate effect by written notice to the other party (the " Breaching Party ") on the occurrence of an event specified in clause 12.2.	14.1			The User shall not assign, novate, declare a trust of or otherwise dispose of this Agreement, or any part thereof, without the prior written approval of Betfair.
12.2	The events referred to in clause 12.1 are:	14.2			The User shall be liable to Betfair for the performance of the User's obligations under this Agreement and for the acts and omissions of its sub-contractors, and where the context requires, references to "the User" in this Agreement shall also include any relevant "sub-contractor".
(a)	the Breaching Party committing an irremediable material breach of a material obligation under this Agreement;	15.			ENTIRE AGREEMENT
(b)	the Breaching Party committing a remediable material breach of a material obligation under this Agreement and failing to remedy the breach within 10 Business Days of the Initiating Party giving reasonable details of the breach and requiring the Breaching Party to remedy such breach;				Subject to clauses 9.2 and 2.4, this Agreement constitutes the entire and only agreement between the parties with regards to its subject matter and each party confirms that it has not been induced to enter into this Agreement in reliance upon, nor has it been given, any warranty (including in particular any warranty as to merchantability,
(c)	the Breaching Party goes into liquidation, either compulsorily or voluntarily or administration or a receiver, administrative receiver, receiver, manager or similar officer is appointed in respect of the whole or any				

fitness for purpose or uninterrupted functionality), representation, statement, assurance, covenant, agreement, undertaking, indemnity or commitment of any nature whatsoever other than as are expressly set out in this Agreement and, to the extent that it has been, it unconditionally and irrevocably waives any claims, rights or remedies which it might otherwise have had in relation thereto.

16. **CUMULATION OF REMEDIES**

Subject to the specific limitations set out in this Agreement, no remedy conferred by any provision of this Agreement is intended to be exclusive of any other remedy except as expressly provided for in this Agreement and each and every remedy shall be cumulative and shall be in addition to every other remedy given thereunder or existing at law or in equity, by statute or otherwise.

17. **NO PARTNERSHIP**

Nothing in this Agreement and no action taken by the parties pursuant to this Agreement shall constitute, or be deemed to constitute, the parties as a partnership, association, joint venture or other co-operative entity.

18. **WAIVER**

18.1 No breach of any provision of this agreement shall be waived or discharged except with the express written consent of the parties.

18.2 No failure or delay by a party to exercise any of its rights under this agreement shall operate as a waiver thereof and no single or partial exercise of any such right shall prevent any other or further exercise of that or any other right.

19. **INVALIDITY AND SEVERABILITY**

19.1 If any provision of this Agreement is or becomes (whether or not pursuant to any judgment or otherwise) invalid, illegal or unenforceable in any respect under the law of any jurisdiction:

- (a) the validity, legality and enforceability under the law of that jurisdiction of any other provision; and
- (b) the validity, legality and enforceability under the law of any other jurisdiction of that or any other provision,

shall not be affected or impaired in any way thereby.

19.2 If any provision of this Agreement shall be held to be void or declared illegal, invalid or unenforceable for any reason whatsoever, such provision shall be divisible from this Agreement and shall be deemed to be deleted from this Agreement and the validity of the remaining provisions shall not be affected. In the event that any such deletion materially affects the interpretation of this Agreement then the parties shall negotiate in good faith with a view to agreeing a substitute provision which as closely as possible reflects the commercial intention of the parties.

20. **NOTICES**

20.1 Notices and communications shall be considered given or made:

- (a) where personally delivered, upon delivery at the address of the relevant party;
- (b) where sent by first class post, three Business Days after the date of posting;
- (c) where sent by air mail, five Business Days after the date of posting;

where delivered by facsimile or email, at the time of transmission, provided that a confirming copy is sent by first class post to the other party within 24 hours after transmission.

21. **THIRD PARTY RIGHTS**

The Contracts (Rights of Third Parties) Act 1999 shall not apply to this Agreement and no rights or benefits expressly or impliedly

conferred by it shall be enforceable under that Act against the parties to it by any other person.

22. **FURTHER ASSURANCE**

Each party shall, upon request from the other, do and execute, or procure that there shall be done and executed, all such documents, deeds, matters, acts or things as that other may at any time require to give it the full benefit of this Agreement.

23. **GOVERNING LAW AND JURISDICTION**

23.1 This Agreement (and any dispute, controversy, proceedings or claim of whatever nature arising out of or in any way relating to this Agreement or its formation) shall be governed by and construed in accordance with English law and the parties hereby irrevocably submit to the jurisdiction of the courts of England and Wales.

24. **EXECUTION AND CHANGES TO AGREEMENT**

24.1 In order to be binding the pricing terms must be agreed as between the parties and inserted into the Term Sheet and then the Agreement must be printed by the user, signed by the user and sent to Betfair for final sign off at Waterfront, Hammersmith Embankment, Chancellors Road, Hammersmith, London W6 9HP. Any changes to the terms of this Agreement must be in writing.

**SCHEDULE 1
Read Only Access**

Read-only
Login
KeepAlive
GetActiveEventTypes
GetAllEventTypes
GetEvents
GetMarket
GetMarketPrices
GetMarketPricesCompressed
GetBetHistory
GetAccountFunds
GetAvailableMktDepth
GetSubscriptionInfo
GetAccountStatement
GetCurrentBets
GetBet
GetMarketProfitAndLoss
GetMarketTradedVolume
GetAllCurrencies
ConvertCurrency

**SCHEDULE 2
Transactional Access**

Transactional (Bet Placement)
Login
KeepAlive
GetActiveEventTypes
GetAllEventTypes
GetEvents
GetMarket
GetMarketPrices
GetMarketPricesCompressed
GetBetHistory
GetAccountFunds
GetAvailableMktDepth
GetSubscriptionInfo
GetAccountStatement
GetCurrentBets
GetBet
GetMarketProfitAndLoss
GetMarketTradedVolume
GetAllCurrencies
ConvertCurrency
PlaceBets
UpdateBets
CancelBets

Reproduced as at Aug 3rd 2006 and may be subject to variation as between individual licensees.

SOAP API Reference Guide

Copyright and Terms	ii
Part I: Betfair API Introduction	9
Chapter 2: Overview	10
WSDL Location.....	10
The Connection End-point URLs for the API Services.....	10
The Architecture of API 5.x	11
What's New in Release 5.2.2.....	16
Where to Find More Information	17
Chapter 3: Session Management.....	18
Betfair's Load Balancing of Client Requests	18
Sessions.....	18
Part II: General API Services Reference.....	20
Chapter 5: API Request/Response Header	21
Chapter 6: Login (global)	23
Chapter 7: Logout (global)	25
Chapter 8: KeepAlive (global).....	26
Part III: Read-Only Betting API Services Reference.....	27
Chapter 9: Get Active Event Types (global)	28
Chapter 10: Get All Event Types (global)	30
Chapter 11: Get All Markets (exchange)	32
Chapter 12: Get Events (global).....	35
Chapter 13: Get Market (exchange)	40
Chapter 14: Get Market Prices (exchange).....	44
Chapter 15: Get Market Prices Compressed (exchange).....	47
Chapter 16: Get Detail Available Market Depth (exchange)	50
Chapter 17: Get Current Bets (exchange).....	52
Chapter 18: Get Matched and Unmatched Bets (exchange)	56
Chapter 19: Get Bet History (exchange)	59
Chapter 20: Get Bet (exchange)	63
Chapter 21: Get Market Profit And Loss (exchange)	66
Chapter 22: Get Market Traded Volume (exchange)	69
Chapter 23: Get All Currencies (global)	71
Chapter 24: Convert Currency (global)	72
Part IV: Bet Placement API Services Reference	73
Chapter 25: Place Bets (exchange).....	74
Chapter 26: Update Bets (exchange)	78
Chapter 27: Cancel Bets (exchange)	82
Part V: Account Management API Services Reference	85
Chapter 28: Create Account (global).....	86

Chapter 29: Retrieve LIMB Message (global)	91
Chapter 30: Submit LIMB Message (global)	95
Chapter 31: View Profile (global)	101
Chapter 32: Modify Profile (global)	103
Chapter 33: Get Account Funds (exchange)	105
Chapter 34: Modify Password (global)	107
Chapter 35: Forgot Password (global)	109
Chapter 36: Get Account Statement (exchange)	111
Chapter 37: Get Subscription Info (global)	114
Chapter 38: Add Payment Card (global)	116
Chapter 39: Get Payment Card (global)	120
Chapter 40: Delete Payment Card (global)	122
Chapter 41: Update Payment Card (global)	124
Chapter 42: Deposit From Payment Card (global)	127
Chapter 43: Withdraw To Payment Card (global)	129
Chapter 44: Transfer Funds (global)	131
Part VI: Appendix	133
Chapter 45: Betfair Simple Data Types	134
AccountStatementEnum	134
AccountStatementIncludeEnum	134
AccountStatusEnum	134
AccountTypeEnum	135
BetsOrderByEnum.....	135
BetStatusEnum	136
BetTypeEnum	136
BillingPeriodEnum	136
CardTypeEnum.....	137
GamcareLimitFreqEnum	137
GenderEnum	137
MarketStatusEnum	138
MarketTypeEnum	138
MarketTypeVariantEnum	138
PaymentCardStatusEnum	139
RegionEnum	139
SecurityQuestion1Enum	139
SecurityQuestion2Enum	140
ServiceEnum.....	140
SortOrderEnum.....	141
SubscriptionStatusEnum.....	141
TitleEnum	142
ValidationErrorsEnum.....	142
Chapter 46: Additional Information	144
Time zones.....	144
Currency	144
Line and Range Markets.....	144

Timestamps.....144
Horse Racing Non-Runners144
Betting In-Running145
Locale Specification145
Part VII: Index.....147
Chapter 47: Index.....148

Part I

Betfair API Introduction

Chapter 2 Overview

This document outlines the services that are available via the Betfair Application Programmers Interface (API) 5.2.2. 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 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://uk.api.sports.betfair.com/exchange/v3/BFExchangeService.wsdl>
- Aus. Exchange: <https://au.api.sports.betfair.com/exchange/v3/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.

This division of the services into global and exchange services is a consequence of the changes that have been made to the architecture of the API in release 5.0. For more information about these changes, see “The Architecture of API 5.x” on page 11.

All the services available from the API are described in the following pages of this developers’ 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://uk.api.sports.betfair.com/exchange/v3/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://au.api.sports.betfair.com/exchange/v3/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).

The Architecture of API 5.x

In release 5.0 Betfair have made certain changes to the architecture of the API. The rest of this chapter explains what these changes are, why they have been made, and what difference they make to you when you use the API.

The Requirements of the Tasmanian Gaming Commission (TGC)

The changes to the API in this release have been made to comply with gaming legislation in Australia. The main requirements of this legislation in its application to the services provided by Betfair 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 New Exchange Servers and Three New URLs

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://au.api.sports.betfair.com/exchange/v3/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://uk.api.sports.betfair.com/exchange/v3/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, using the session token you received in the previous API response header.

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.

For release 5.0, 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 "Which are the "Global" and Which the "Exchange" Services?" on page 12.

Finally, this distinction between the two types of services available is also now reflected in the provision of two distinct WSDL files for the API: there is one for the global services and one for the exchange services.

Which are the "Global" and Which the "Exchange" Services?

The global services available are listed below:

- `addPaymentCard`
- `convertCurrency`
- `createAccount`
- `deletePaymentCard`
- `depositFromPaymentCard`
- `forgotPassword`
- `getActiveEventTypes`
- `getAllCurrencies`
- `getAllEventTypes`
- `getEvents`
- `getPaymentCard`
- `getSubscriptionInfo`
- `keepAlive`
- `login`
- `logout`
- `modifyPassword`
- `modifyProfile`
- `retrieveLIMBMessage`
- `submitLIMBMessage`
- `transferFunds`
- `updatePaymentCard`
- `viewProfile`
- `withdrawToPaymentCard`

The exchange services available from the exchange specific services are listed below:

- cancelBets
- getAccountFunds
- getAccountStatement
- getAllMarkets
- getBet
- getBetHistory
- getCurrentBets
- getDetailAvailableMktDepth
- getMUBets
- getMarket
- getMarketPrices
- getMarketPricesCompressed
- getMarketProfitAndLoss
- getMarketTradedVolume
- placeBets
- updateBets

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 - in the sense in which the word is used by Betfair - is really a data structure that is stored locally to an exchange server and that 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 in question, 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 on are lifted from a customer’s Australian wallet.

New Services in API 5.0 and Changes to Existing Services

The following table shows the services that have changed between API 4.1.0 and API 5.0.

Table 2-1: New Services and Key Changes to Existing Services

API 5.0 Service	Description
getEvents	<p>Allows you to request the lists of events that it is possible to bet on. From release 5.0 onwards of the API, the betting market data for an event returned by the getEvents service will be available from one of two exchange servers. This means that, to retrieve market data (for example, using the getMarket service), you need to direct your client software to the correct exchange server for that event.</p> <p>To enable you to decide which exchange server to send a getMarket request to, the getEvents service has a new output parameter called “exchangeId”. This will have the value ‘1’ for the UK exchange server and ‘2’ for the Australian exchange server.</p> <p>For the correct URL to use to contact the exchange server indicated by the exchangeId in a getEvents response, see the entry in this table (below) for getMarket.</p> <p>In API 5.0 a new output parameter called CouponLinks is returned. This is a list of any Betfair coupons that include the market or markets associated with a particular event. Please note that this data is designed for use with a service to be implemented in a future release of the API (a service for requesting specific coupons). Until then the data is redundant.</p> <p>Also in API 5.0 a new output parameter called numberOfWinners has been added to the GetEvents service. This tells you the number of winning selections that are possible in the betting market for an event. It helps you determine whether the market is a win or place market. If the value is ‘1’, the market is a win market; if the value is ‘2’, the market is a place market.</p> <p>Finally, from release 5.0 of the API, the getEvents service returns details of line and range markets, where these markets are available.</p> <p>For more information about the getEvents service, see Chapter 12.</p>
getMarket	<p>Allows you to retrieve betting market data for an event. From release 5.0 onwards of the API, betting markets for events are distributed between two exchange servers, and you must send your getMarket request to the correct server for the event you are interested in (see getEvents above).</p> <p>Depending on the “exchangeId” value returned by getEvents, the end-point URLs for a getMarket request are:</p> <p>https://api.betfair.com/exchange/v3/BFExchangeService (this is for UK and worldwide sports events, excluding Australian ones),</p> <p>https://api-au.betfair.com/exchange/v3/BFExchangeService (this is for Australian events).</p> <p>Note that if you send a getMarket request to the Australian exchange server when the market you are interested in is on the UK exchange server (or vice versa) you will receive an error message. The exchange servers do not search each other for market data (the geographical distance between them forbids this). However, note that it is now possible for a marketId on one exchange server to be duplicated on another server. This makes it even more important that you send your service requests (especially betting requests) to the correct server.</p> <p>The API’s response to a getMarket request now includes a “licenceId” parameter. This indicates the licence under which the betting market is operating. It will have the value ‘1’ for a UK licence and ‘2’ for an Australian licence.</p>

Table 2-1: New Services and Key Changes to Existing Services

API 5.0 Service	Description
getMarket (cont'd)	<p>The API now requires a (Boolean) input parameter called includeCouponLinks to be submitted in requests for getMarket. If you set this parameter to true, the getMarket response will contain a list of any current Betfair coupons that include the market you have requested.</p> <p>Although includeCouponsLink is a mandatory input parameter, it is not of immediate use. If you set it to true, it will return coupon information (listing any Betfair coupons that include the market that has been returned in the getMarket response). However, the includeCouponsLink parameter is designed for use in a future implementation of the API that will include a service enabling you to request coupons. Until this service is available, the data returned by includeCouponsLink is redundant. (If you set includeCouponLinks to false, no coupon data will be returned.)</p> <p>You will notice that the eventHierarchy array returned by the getMarket request (see Table 13-3 on page 41) is empty in output received from the Australian exchange server. 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. This history is stored on the database systems that support the API's global services, and since these databases are located in Europe they are not available to the Australian exchange server. Therefore, when you send a getMarket request to the Australian exchange server, the eventHierarchy is an unpopulated array.</p> <p>Finally, from release 5.0 of the API, the getMarket service returns details of line and range markets, where these markets are available. Four new output parameters are provided in this connection: unit, minUnitValue, maxUnitValue, and interval (see Table 13-3 on page 41). For more information, see Chapter 13.</p>
transferFunds	<p>This is a new service that has been added in API 5.0. It allows you to transfer funds from your main Betfair betting wallet into a dedicated wallet for betting on sports events taking place in Australia. It also allows you to transfer funds from your Australian wallet into your main Betfair wallet so that you can use them to bet on other events or withdraw them to your bank account. For more information, refer to Chapter 44.</p>
createAccount	<p>Allows Betfair partners to create and modify user accounts. (Please note that this service is available only to selected Betfair partners.)</p> <p>From release 5.0 of the API, the createAccount service creates two local wallets for each user account created. One is for the UK exchange server, and one is for the Australian exchange server. The UK wallet is the master wallet: it is this one that you can pay money into and take money out of by means of payment cards or other methods. The Australian wallet is only for betting on events whose betting market is hosted on the Australian exchange server. It is a subordinate wallet: you cannot pay money directly into it or withdraw money from it; you can only transfer funds into it from the UK wallet or out of it to the UK wallet by means of the transferFunds service (described above in this table).</p> <p>This service is affected by a new requirement for Betfair users whose country of residence is Australia: they must submit the name of their state and not their county as the value of the input parameter "countyState" (see Chapter 28).</p> <p>Finally, a new input parameter has been added for this service. This is the preferredName parameter, which specifies the name that a user prefers to be addressed by in correspondence with Betfair.</p>
modifyProfile	<p>This service is affected by the same requirement as the createAccount service for Australian residents to submit the name of their state and not their county as the value of the input parameter "countyState" (see Chapter 32).</p> <p>A new input parameter has been added for this service. This is the manualAddress parameter. It indicates whether the user's address has been entered manually by the user or has been generated entirely or in part by some automated process.</p>

Table 2-1: New Services and Key Changes to Existing Services

API 5.0 Service	Description
getAccountFunds	Allows you to check the balance of your local wallet on a particular exchange server. This service existed in releases of the API preceding 5.0. However, from release 5.0 onwards, its function is slightly different: it used to tell you the balance of your account; but now it tells you the balance of your wallet on a particular exchange server (the exchange server that you send the service request to). For more information, see Chapter 33.
getAccountStatement	Allows you to view the transactions that have taken place in your local wallet on a particular exchange server. This service existed in releases of the API preceding 5.0. However, from release 5.0 onwards, its function is slightly different: it used to list all the transactions that had taken place in your account; but now it lists only the transactions affecting your wallet on a particular exchange server (the exchange server that you send the service request to). For more information, see Chapter 36.
getBetHistory, getCurrentBets, getMUBets	<p>Now that there are two betting exchange servers, the information that these API services return concerns only the bets you have placed on the exchange server you sent the request to. For information about bets on the other exchange server, you must send the requests to the other exchange server.</p> <p>In addition the following changes have been made in API 5.0:</p> <ul style="list-style-type: none"> ■ A betIds input parameter now exists for getMUBets. This enables you to retrieve specific matched or unmatched bets (on a particular exchange server) by their betId. ■ A matchedSince input parameter now exists for getMUBets. This enables you to retrieve bets matched since a specified date and time (it can also optionally return your unmatched bets - since the specified date - along with the matched ones). ■ A marketId input parameter now exists for getCurrentBets. This enables you to retrieve all your bets placed in a specific market. ■ A noTotalRecordCount parameter now exists for getCurrentBets. This enables you to specify that you do not want the API to perform a count of the records it is returning. (Performing the record count slows down the API's response to your request.)
getActiveEventTypes, getAllEventTypes	New output parameters are returned in each eventTypeItems array that is returned by these services. If a marketId is present in the nextMarketId parameter for an eventType, then the new exchangeId parameter indicates which exchange server is hosting that market. The possible values for exchangeId are '1' for the UK exchange server, and '2' for the Australian exchange server. (For more information, see Chapter 9 and Chapter 10.)
addPaymentCard	<p>Three new input parameters have been added to the addPaymentCard request: these are the town and country parameters and the issueNumber parameter. The town parameter specifies the cardholder's town of residence; the county parameter specifies his or her county of residence. The issueNumber is for Switch and Solo cards: it specifies their issue number (its value needs to be set to NULL for all other cards).</p> <p>The billingName and county have also been added as parameters to the output for this service (town was already returned in the output and continues to be so).</p>
getPaymentCards	The cardholder's billingName and county have been added to the output parameters provided by this service.
updatePaymentCard	The issueNumber, town, county, and cardStatus parameters have been added to the list of input parameters for this service.

What's New in Release 5.2.2

This release includes the following changes:

- New wsdl and endpoint URLs for accessing the Betfair API. The existing URLs can still be used and will be available until four month past the release date (Dec 06), however you should convert your software to use the new URLs as soon as possible.
- An update to the [GetAllMarkets](#) service to properly escape : (colon) characters that appear in market and event names.
- The restriction on cancelling bets from Australia while the market is in-play has been removed to maintain consistency with the website.
- An update to the [GetBetHistory](#) service to return cancelled bets for only settled markets to maintain consistency with the website.

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 2-2: More Information

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

Chapter 3 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 and cookies

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.

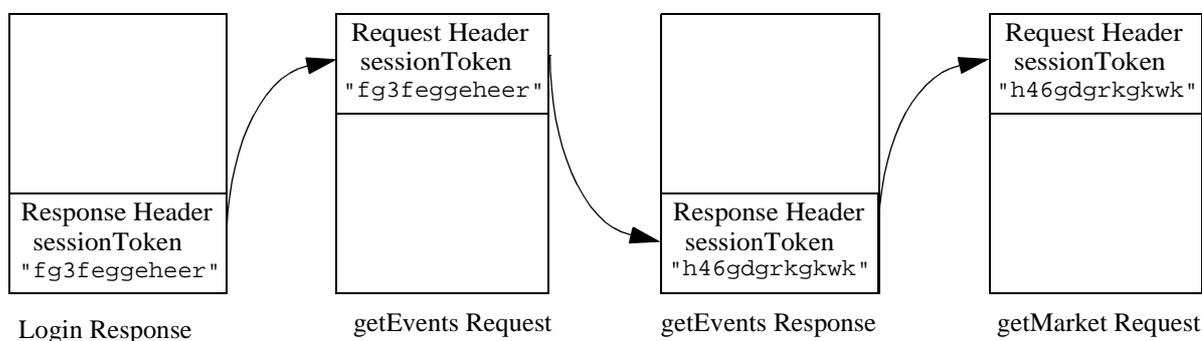
There is one load balancer for the global services, one for the UK exchange server, and one for the Australian exchange server. In each of these cases, the load balancer distributes service requests across the available application servers.

The Betfair API uses two mechanisms to maintain state for an API client:

- The `sessionToken` is a mandatory field in the API Request header used to track log-ins.
- The API cookie is used to route requests to API servers. The cookie is not mandatory, but for best performance it is highly recommended that you store this cookie and return it each time you request a service.

Sessions

After a successful log-in the Betfair API returns a `sessionToken` in its response. The `sessionToken` is used to maintain state information. It acts as a key proving that a successful login has been performed. A `sessionToken` can only be used for a single request; the response to the next request returns another token, and that token must be used for the next request after that. (The diagram below illustrates this sequence.) Each time you make an API service request, whether for a global or an exchange service, you must include the `sessionToken` that was returned in the last service response.



Note: As mentioned in Chapter 1, 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 URLs. (For information about the distinction between global and exchange services see Chapter 2.)

The following example shows an example XML request and response for the [login](#) service. The request is:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAPENV=" http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
www.w3.org/1999/XMLSchema" xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance">

<SOAP-ENV:Body>
<login xmlns="http://www.betfair.com/publicapi/BFService/">
<request>
<username>AUserName</username>
<password>APassWord</password>
<productId>22</productId>
<vendorSoftwareId>0</vendorSoftwareId>
<locationId>0</locationId>
</request>
</login>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The response received is:

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

<soap:Body>
<n:loginResponse xmlns:n="http://www.betfair.com/publicapi/BFService/">
<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">r6n+yBb0nm3ret5rkj4ti4Zg</sessionToken>
<timestamp xsi:type="xsd:dateTime">2006-01-06T13:27:58.330Z</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">2006-02-05T00:00:00.000Z</validUntil>
</n:Result>
</n:loginResponse>
</soap:Body>
</soap:Envelope>
```

In this example, the [sessionToken](#) contains the value `r6n+yBb0nm3ret5rkj4ti4Zg`.

Multiple Sessions

It is possible to use multiple sessions concurrently (such as for a multi-threaded application). This can be implemented by performing multiple log ins, therefore creating multiple sessions and receiving multiple session tokens.

If you want to make multiple request in from your application, you must call [login](#) for each stream of requests, and track the session token values returned for each login.

Part II

General API Services Reference

Chapter 5 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.

The following table describes the `APIRequestHeader`.

Table 1: APIRequestHeader

Parameter	Type	Description
<code>clientStamp</code>	Long	Not currently used and should be set to zero
<code>sessionToken</code>	String	Session Identifier

The following table describes the `APIReponseHeader`.

Table 2: APIResponseHeader

Parameter	Type	Description
<code>errorCode</code>	APIErrorEnum	If not null, indicates a non service specific error has occurred
<code>minorErrorCode</code>	String	Reserved for future use - currently always null
<code>sessionToken</code>	String	Uniquely identifier for next request in this session. This token must be set in the <code>APIRequestHeader</code> for the next service invoked.
<code>timestamp</code>	DateTime	The time at which the response was returned from the server

The following table describes the possible error codes returned by the Betfair API.

Table 3: 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 has expired

Table 3: APIErrorEnum

Value	Description
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 account has a positive balance.
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.

Chapter 6

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: 1 instance of LoginReq

Parameter	Mandatory?	Type	Description
username	Y	String	The username with which to login to the API for a new session. The username must be at least 6 characters and no longer than 20 characters.
password	Y	String	The password with which to login to the API for a new session. The password must be at least 8 characters and no longer than 20 characters
productId	Y	Integer	The API product ID with which to login to the API for a new session. This is provided when you sign 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

Output

The following table describes the returned parameters for **Login**.

Table 6-2: 1 instance of LoginResp

Parameter	Type	Notes
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
currency	String	The currency used by the account
errorCode	LoginErrorEnum	login specific error code (see enum below)
minorErrorCode	String	Reserved for future use - currently always null
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**.

Table 6-3: LoginErrorEnum

Value	Condition
OK	
ACCOUNT_CLOSED	Account closed - please contact BDP support.
ACCOUNT_SUSPENDED	Account suspended - please contact BDP support.
API_ERROR	General API Error
FAILED_MESSAGE	The user cannot login until they acknowledge a message from Betfair (see Chapter 29).
INVALID_LOCATION	Invalid LocationID
INVALID_PRODUCT	Invalid productID entered
INVALID_USERNAME_OR_PASSWORD	Incorrect username and/or password supplied.
INVALID_VENDOR_SOFTWARE_ID	Invalid vendorSoftwareId supplied
LOGIN_FAILED_ACCOUNT_LOCKED	Account locked due to too many failed login attempts
LOGIN_REQUIRE_TERMS_AND_CONDITIONS_ACCEPTANCE	User has new T&C to agree to. You can either use the RetrieveLIMBMessage and SubmitLIMBMessage services or accept the new terms through the Betfair website. In addition, if you do nothing your continued usage of the Betfair API and/or website will be considered acceptance of the new terms.
LOGIN_RESTRICTED_LOCATION	Login origin from a restricted country
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
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	Account locked, Please login to the Betfair website and assent to the terms and conditions.
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. Field name that is passed to subsequent calls is now called sessionToken.

Chapter 7 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 7-1: 1 instance of `LogoutReq`

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).

Output

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

Table 7-2: 1 Instance of `LogoutResp`

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
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 7-3: `LogoutErrorEnum`

Value	Condition
OK	
API_ERROR	General API Error

Chapter 8 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. The `sessionToken` returned by the Betfair API changes for each call. Every time your application calls the Betfair API and is returned a new `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 parameters used for calling the `KeepAlive` service.

Table 8-1: 1 instance of `KeepAliveRequest`

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Response Header that contains the user's session token (see Chapter 5).

Output

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

Table 8-2: 1 instance of `KeepAliveResp`

Parameter	Type	Notes
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
apiVersion	String	The current version of the API
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.

Part III

Read-Only Betting API Services Reference

Chapter 9 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 9-1: 1 instance of [GetActiveEventTypesReq](#)

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Response Header that contains the user's session token (see Chapter 5).
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 [GetActiveEventTypes](#) service.

Table 9-2: 1 Instance of [GetEventTypesResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
eventTypeItems	ArrayOfEventType	An array containing the active event types
minorErrorCode	String	Reserved for future use - currently always null
errorCode	GetEventsErrorEnum	Service specific error code

The [GetActiveEventTypes](#) service returns an array of [eventTypeItems](#). The table below shows the fields in [EventType](#).

Table 9-3: [ArrayOfEventType](#) 0 or more instances of [EventType](#)

Parameter	Type	Description
id	Integer	The ID of Event

Table 9-3: ArrayOfEventType 0 or more instances of EventType

Parameter	Type	Description
name	String	The name of the Event Type
exchangeId	Integer	<p>The Id of the exchange server that is hosting the betting markets for the eventType.</p> <p>From release 5.0 onwards of the API, the betting market data for an event 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 eventType. For this reason the getActiveEventTypes 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 for an event are:</p> <p>https://api.betfair.com/exchange/v3/BFExchangeService (this is for UK and worldwide sports events, excluding Australian ones),</p> <p>https://api-au.betfair.com/exchange/v3/BFExchangeService (this is for Australian events).</p> <p>For information about using the getMarket service, see Chapter 13.</p> <p>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 44.</p>
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 9-4: GetEventsErrorEnum

Value	Condition
OK	
API_ERROR	General API Error
INVALID_EVENT_ID	Not used
INVALID_LOCALE_DEFAULTING_TO_ENGLISH	The locale string was not recognized. Returned results are in English.
NO_RESULTS	There are no events to return

Chapter 10

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 10-1: 1 instance of [GetEventTypesReq](#)

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
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 10-2: 1 Instance of [GetEventTypesResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
eventTypeItems	ArrayOfEventType	Array of event types
minorErrorCode	String	Reserved for future use - currently always null
errorCode	GetEventsErrorEnum	Service specific error codes. See below

The [GetAllEventTypes](#) service returns [eventTypeItems](#) as an array. The table below shows the fields in [EventType](#)

Table 10-3: [ArrayOfEventType](#) 0 or more instances of [EventType](#)

Parameter	Type	Description
id	Integer	The ID of Event

Table 10-3: ArrayOfEventType 0 or more instances of EventType

Parameter	Type	Description
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 nextMarketId parameter, then the exchangeId parameter indicates the exchange server that is hosting that market. The possible values for the exchangeId 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 10-4: GetEventsErrorEnum

Value	Condition
OK	
API_ERROR	General API Error
INVALID_EVENT_ID	Not used
INVALID_LOCAL_DEFAULTING_TO_ENGLISH	The locale string was not recognized. Returned results are in English.
NO_RESULTS	No data available to return

Chapter 11 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.

Tip: 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](#)

Input

The following table describes the parameters used for calling the [GetAllMarkets](#) service.

Table 11-1: 1 instance of [GetAllMarketsReq](#)

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token. See "API Request/Response Header" on page 21.
locale	N	String	The locale to use when returning results. If not specified, the default locale for the user's account is used.
eventTypesIds	N	ArrayOfInt	The events types to return. If not specified, markets from all event types are returned.
countries	N	ArrayOfCountryCode	The countries where the event is taking place as an array of ISO3 country codes. If not specified, markets from all countries 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.
toDate	N	DateTime	If this is set, the response contains only markets where the market time is not after the specified date.

Output

The following table describes the parameters returned from the [GetAllMarkets](#) service.

Table 11-2: 1 Instance of [GetAllMarketsResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token. See "API Request/Response Header" on page 21.
errorCode	GetAllMarketsErrorEnum	Service specific error code

Table 11-2: 1 Instance of `GetAllMarketsResp`

Parameter	Type	Description
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 11-3: 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 Event Type Name at the beginning or 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. Unlike the menu path, this includes the Event Type Id and 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
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.

Example 11-1: Example marketData output

```
:20158165~Match Odds~0~ACTIVE~1164223800000~\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:
```

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...

Tip: 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 11-4: GetAllMarketsErrorEnum

Value	Condition
OK	
API_ERROR	General API Error
INVALID_COUNTRY_CODE	The country code submitted does not match a known country code.
INVALID_EVENT_TYPE_ID	The eventId does not match any known eventTypes. See “Get All Event Types (global)” on page 30
INVALID_LOCALE	The locale specified does not exist. See “Locale Specification” on page 145

Chapter 12

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 (see Table 12-5 on page 37). 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 13.

Warning: 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). 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 13.

Table 12-1: 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 12-2: 1 instance of [GetEventsReq](#)

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
eventParentId	Y	Integer	This is either an Id value for a single item (in an array of eventTypeIdItems 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 12-3: 1 Instance of [GetEventsResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
errorCode	GetEventsErrorEnum	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

Table 12-3: 1 Instance of `GetEventsResp`

Parameter	Type	Description
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 12-4: `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.
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 12-5: `ArrayOfMarketSummary` 0 or more instances of `MarketSummary`

Parameter	Type	Description
eventTypeid	Integer	The Event Type of the market

Table 12-5: ArrayOfMarketSummary 0 or more instances of MarketSummary

Parameter	Type	Description
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 13. 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 44.
eventParentId	Integer	This is either an Event ID or an Event Type ID
marketId	Integer	The ID of the market
marketName	String	The name of the market
marketType	MarketTypeEnum	The market type (e.g. Asian Handicap, Odds). For more information, see Chapter 45.
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 12-6: 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 12-7: GetEventsErrorEnum

Value	Condition
OK	
API_ERROR	General API Error
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.

Chapter 13

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`

Input

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

Table 13-1: 1 instance of `GetMarketReq`

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
marketId	Y	Integer	The market ID.
includeCouponLink	Y	Boolean	If you set this parameter to true, the service response will contain a list of any coupons that include the market you have requested. Although the parameter is mandatory, it is of no practical value in the current release of the API. Future implementations of the API will include a service that enables you to request the coupons whose details have been returned to you in a <code>GetMarket</code> response. Until this is available, however, the data is redundant. If you set the parameter to false, no coupon data will be 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 13-2: 1 Instance of `GetMarketResp`

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
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 an array called [Market](#). The table below shows the fields in a [Market](#)

Table 13-3: Market

Parameter	Type	Description
countryISO3	ISO3 String	
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.
eventTypeid	Integer	The event type id of the market (previously sportId). Not currently used.
lastRefresh	Long	The time the market information was last read from the database expressed as the number of milliseconds since Jan. 1, 1970 GMT.
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 45.
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 45.
marketTypeVariant	MarketTypeVariantEnum	The Market variant for Asian Handicap markets. For more information, see Chapter 45.
menuPath	String	The detailed path through the Betfair menu to reach this market
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.
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)

Table 13-3: Market

Parameter	Type	Description
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.
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.
minUnitValue	Integer	The minimum unit value (for example, the minimum number of goals) that bets can be matched against in this market.
maxUnitValue	Integer	The maximum unit value (for example, the maximum number of goals) that bets can be matched against in this 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 with “interval”; it has the same meaning.)
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
couponLinks	arrayOfCouponLinks	A list of the coupons that include this market.

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 13-4: 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 13-5: 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 13-6: GetMarketErrorEnum

Value	Condition
OK	
API_ERROR	General API Error
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. Currently, the API does not support Line and Range markets.

Chapter 14 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`

Input

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

Table 14-1: 1 instance of `GetMarketPricesReq`

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
currencyCode	Y	String	Three letter ISO 4217 code. If not supplied, users currency is used
marketId	Y	Integer	ID of the required market

Output

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

Table 14-2: 1 Instance of `GetMarketPricesResp`

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
errorCode	MarketPricesErrorEnum	getMarketPrices specific error code (see enum below)
marketPrices	MarketPrices	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 14-3: `MarketPrices`

Parameter	Type	Description
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	The time the market information was last read from the database
marketBaseRate	Float	This will indicate the commission price set for each market.

Table 14-3: MarketPrices

Parameter	Type	Description
marketId	Integer	ID of the market
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 45.
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	ArrayOfRunnerPrices	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 14-4: ArrayOfRunnerPrices

Parameter	Type	Description
asianLineId	Integer	Id of the selection (this will be the same for the same selection across markets)
bestPricesToBack	ArrayOfPrice	Best available back prices
bestPricesToLay	ArrayOfPrice	Best available lay prices
handicap	Double	Handicap of the market (applicable to Asian handicap markets)
lastPriceMatched	Double	The last price at which this selection was matched
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 14-5: ArrayOfPrice

Parameter	Type	Description
amountAvailable	Double	Amount available at the odds specified.

Table 14-5: ArrayOfPrice

Parameter	Type	Description
betType	BetTypeEnum	Bet type data. For more information, see Chapter 45.
depth	Integer	The order, from best to worst, of the price ('1' is best available)
price	Double	Odds

Error Code

The following table describes the error codes returned by [GetMarketPrices](#).

Table 14-6: MarketPricesErrorEnum

Value	Condition
OK	
API_ERROR	General API Error
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.

Chapter 15

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`

Input

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

Table 15-1: 1 instance of `GetMarketPricesCompressedReq`

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
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 15-2: 1 Instance of `GetMarketPricesCompressedResp`

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
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 `marketPrices`

Table 15-3: 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

Table 15-3: marketPrices

Field No.	Type	Name	Notes
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?
8	String	Market Base Rate	Base rate of commission on market
9	Long	Refresh time in milliseconds	The time the market information was last read from the database
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	DateTime	Removed Date	The date the runner was removed
3	String	Adjustment Factor	The adjustment 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)
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
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

Table 15-3: marketPrices

Field No.	Type	Name	Notes
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 15-1: 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;:1058616~0~6.04~8.4~11.9~false||:670160~1~6.18~17.5~4.2~false||:1132
008~2~9.78~5.2~20.4~false||:894820~3~140.02~4.6~20.4~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||:1433455~6~0.0~0.4~false||:
1433456~7~0.0~0.9~false||:746753~8~5.54~11.5~5.2~false||:1433457~9~0.0~4.2~fals
e||:1147548~10~0.0~2.6~false||:1433458~11~62.46~2.0~3.5~false||:1433459~12~0.0~
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).

Error Code

The following table describes the error codes returned by [GetMarketPricesCompressed](#).

Table 15-4: MarketPricesErrorEnum

Value	Condition
OK	
API_ERROR	General API Error
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. Currently, this includes Line and Range markets.

Chapter 16

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: The data returned by `GetDetailAvailableMarketDepth` is stored by the exchange server in a cache that is only updated every 20 seconds (approximately). This cache is separate from the one containing the data that is returned by the `GetMarketPrices` service (see Chapter 14), for example, which is updated much more frequently. Do not use the `GetDetailAvailableMarketDepth` service for client application functions that require real-time market data.

Service Details

SOAP Action: `getDetailAvailableMktDepth`

Input

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

Table 16-1: 1 instance of `GetDetailedAvailMktDepthReq`

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
currencyCode	Y	String	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
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 16-2: 1 Instance of `GetDetailedAvailMktDepthResp`

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
errorCode	GetDetailedAvailMktDepthErrorEnum	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 16-3: ArrayOfAvailabilityInfo 0 or more instances of AvailabilityInfo

Parameter	Type	Description
odds	Decimal	The price of the bet
totalAvailableBackAmount	Decimal	Total amount available to lay at the given odds
totalAvailableLayAmount	Decimal	Total amount layed at the given odds

Error Code

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

Table 16-4: GetDetailedAvailMktDepthErrorEnum

Value	Condition
OK	
API_ERROR	General API Error
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. It may be a Line or Range market.
NO_RESULTS	No results were returned for the request arguments
SUSPENDED_MARKET	The market Id specified refers to a suspended market

Chapter 17 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.

Service Details

SOAP Action: `getCurrentBets(for UK/worldwide events)`

Input

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

Table 17-1: 1 instance of `GetCurrentBetsReq`

Parameter	Mandatory?	Type	Description
<code>header</code>	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
<code>betStatus</code>	Y	BetStatusEnum	If <code>MarketID</code> is supplied, bet status must be U, M, or S - see Chapter 45 for details.
<code>detailed</code>	Y	Boolean	no effect
<code>locale</code>	N	String	Specify the language for the reply if you want a different language than the account default. For more information, see "Locale Specification" on page 145.
<code>timezone</code>	N	String	If a timezone is specified and the <code>fullMarketName</code> field contains an event time, the event time is adjusted to the specified timezone.
<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 server that you send the request to.
<code>orderBy</code>	Y	BetsOrderByEnum	If <code>Ordering</code> is not NONE - please see Chapter 45.
<code>recordCount</code>	Y	Integer	Maximum number of records to return
<code>startRecord</code>	Y	Integer	First record number to return

Table 17-1: 1 instance of [GetCurrentBetsReq](#)

Parameter	Mandatory?	Type	Description
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 (see Table 17-3 on page 53). 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 allowed ordering for each Bet status.

Table 17-2: Allowed Ordering, Bet status combinations

	Unmatched	Matched	Settled	Cancelled	Lapsed	Voided
NONE	Y	Y	Y	Y	Y	Y
BET_ID						
MARKET_NAME						
PLACED_DATE	Y	Y				
MATCHED_DATE		Y				
CANCELLED_DATE				Y		

Output

The following table describes the parameters returned from the [GetCurrentBets](#) service.

Table 17-3: 1 Instance of [GetCurrentBetsResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
bets	ArrayOfBet	The Bet information
errorCode	GetCurrentBetsErrorEnum	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 17-4: 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)

Table 17-4: ArrayOfBet 0 or more instances of Bet

Parameter	Type	Description
betId	Integer	Unique identifier generated for every bet placement
betStatus	BetStatusEnum	Status of the bet - please see Chapter 45.
betType	BetTypeEnum	Back or lay - please see Chapter 45.
cancelledDate	dateTime	Date and time that the bet was cancelled (null if not applicable)
handicap	Double	Handicap of the market (applicable to Asian handicap markets)
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 market name, including the parent event names, up to three levels in the hierarchy
marketType	MarketTypeEnum	Market type data - please see Chapter 45.
marketTypeVariant	MarketTypeVariantEnum	The Market variant for Asian Handicap markets - please see Chapter 45.
matchedDate	dateTime	Date and time that the bet was matched (null if not applicable)
matchedSize	Double	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 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
requestedSize	Double	Original stake amount of the bet
voidedDate	Date	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 17-5: ArrayOfMatch 0 or more instances of Match

Parameter	Type	Description
betStatus	BetStatusEnum	Status of the bet. For more information, please see Chapter 45.
matchedDate	dateTime	Date and time that the bet portion was matched
priceMatched	Double	Price at which this portion was matched

Table 17-5: ArrayOfMatch 0 or more instances of Match

Parameter	Type	Description
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 17-6: GetCurrentBetsErrorEnum

Value	Condition
OK	
API_ERROR	General API Error
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"> Bet Status is MATCHED and Ordering is neither MATCHED_DATE or PLACED_DATE Bet Status is UNMATCHED and ordering isn't PLACED_DATE 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	

Chapter 18

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](#)

Input

The following table describes the parameters used for calling the [GetMUBets](#) service.

Table 18-1: 1 instance of [GetMUBetsReq](#)

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
betStatus	Y	BetStatusEnum	The status of the bets to return (matched, unmatched, or both) - please see Chapter 45.
marketId	N	Integer	Limit the bets returned to those placed on a single market.
betIds	N	ArrayOfInt	Specifies the betId of each bet you want returned. (If the betIds array contains at least one betId, then the marketId parameter will be ignored and only the specified bet or bets returned.) The maximum number of bets you can include in the array is 200.
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	BetsOrderByEnum	The order of returned results. If Ordering is not NONE, please see Chapter 45. Valid orders are BET_ID, PLACED_DATE, and MATCHED_DATE.
sortOrder	Y	SortOrderEnum	Whether the results are in ascending or descending order
recordCount	Y	Integer	Maximum number of records to return
startRecord	Y	Integer	First record number to return

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 18-2: Ordering Combinations by Bet status

Order By	Bet Status					
	Matched		Unmatched		Matched/Unmatched	
	<i>First sorted by...</i>	<i>then by...</i>	<i>First sorted by...</i>	<i>then by...</i>	<i>First sorted by...</i>	<i>then by...</i>
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 18-3: 1 Instance of [GetMUBetsResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
bets	ArrayOfMUBet	The Bet information
errorCode		Specific error code (see enum below)
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 18-4: [ArrayOfMUBet](#) 0 or more instances of [MUBet](#)

Parameter	Type	Description
asianLineId	Integer	Id of the specific Asian handicap line
betId	Integer	Unique identifier generated for every bet placement
betStatus	BetStatusEnum	Status of the bet - please see Chapter 45.
betType	BetTypeEnum	Back or lay - please see Chapter 45.
marketId	Integer	Id of the market
matchedDate	dateTime	Date and time that the bet was matched (null if not applicable)
size	Double	The amount matched
placedDate	dateTime	Date and time of bet placement
price	Double	Price of the remaining bet

Table 18-4: ArrayOfMUBet 0 or more instances of MUBet

Parameter	Type	Description
selectionId	Integer	Id of the selection (this will be the same for the same selection across markets)
executedBy	String	Used for internal Betfair applications.
handicap	Double	Used for internal Betfair applications.

Error Codes

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

Table 18-5: GetMUBetsErrorEnum

Value	Condition
OK	
API_ERROR	General API Error
INVALID_BET_STATUS	Invalid bet status. The bet status must be M, U, or 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"> Bet Status is M (matched) and Ordering is neither MATCHED_DATE or PLACED_DATE 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 invalid
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.

Chapter 19 Get Bet History (exchange)

The API `GetBetHistory` service allows you to retrieve information about the bets you have placed on a particular exchange server. 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.

Service Details

SOAP Action: `getBetHistory`

Input

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

Table 19-1: 1 instance of `GetBetHistoryReq`

Parameter	Mandatory?	Type	Description
<code>header</code>	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
<code>betTypesIncluded</code>	Y	BetStatusEnum	Indicates the status of the bets to include in the response. For a list of the valid values, see Chapter 45. If your <code>betHistory</code> request is for a specific market (in other words, if you have specified a <code>marketId</code> in your request), then you must specify either M or U as the value for <code>betTypesIncluded</code> . Otherwise you will receive an <code>INVALID_BET_STATUS</code> error. Only settled markets return cancelled, void, or lapsed bets.
<code>detailed</code>	Y	Boolean	Whether to show details of all the matches on a single bet
<code>eventTypeIds</code>	Y	ArrayOfInt	Event types to return. If you submit an empty array, you will receive records of all your bets on the exchange that you send the request to (unless you also specify a <code>marketId</code> , in which case, the API will ignore the contents of <code>eventTypeIds</code> array).
<code>marketId</code>	N	Integer	Returns the records of your bets for the specified market. If you use this parameter (in other words, if do specify a <code>marketId</code> in your request), the API will ignore the contents of the <code>eventTypeIds</code> input parameter. Note that, if you specify a <code>marketId</code> , you must also specify either M or U as the value for the <code>betTypesIncluded</code> parameter (see above in this table).

Table 19-1: 1 instance of [GetBetHistoryReq](#)

Parameter	Mandatory?	Type	Description
locale	N	String	Specify the locale for the reply if you want a different language than the account default. For more information, see “Locale Specification” on page 145
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 45.
placedDateFrom	Y	Date	Return records on or after this date. (The date datatype 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 datatype 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 45.
startRecord	Y	Integer	The first record number to return (supports paging)

Output

The following table describes the parameters returned from the [GetBetHistory](#) service.

Table 19-2: 1 Instance of [GetBetHistoryResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
betHistoryItems	ArrayOfBet	See Table 19-3 on page 61.
errorCode	GetBetHistoryErrorEnum	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 19-3: 1 instance of `Bet`

Parameter	Type	Description
<code>asianLineId</code>	Integer	Id of the specific Asian handicap line
<code>avgPrice</code>	Double	The average matched price of the bet (null if no part has been matched)
<code>betId</code>	Integer	Unique identifier generated for every bet placement
<code>betStatus</code>	BetStatusEnum	Status of the bet - please see Chapter 45.
<code>betType</code>	BetTypeEnum	Back or lay - please see Chapter 45.
<code>cancelledDate</code>	dateTime	Date and time that the bet was cancelled (null if not applicable)
<code>lapsedDate</code>	dateTime	Date and time that the bet was lapsed (null if not applicable)
<code>marketId</code>	Integer	Id of the market
<code>marketName</code>	String	Name of the market
<code>fullMarketName</code>	String	The full name of the market. This string is localised to the language set in the <code>locale</code> parameter (if set) otherwise to the user's language as specified in their profile.
<code>marketType</code>	MarketTypeEnum	Market type data - please see Chapter 45.
<code>matchedDate</code>	dateTime	Date and time that the bet was matched (null if not applicable)
<code>matchedSize</code>	Double	The amount matched
<code>matches</code>	ArrayOfMatch	If detailed is true, this will be the details of the matched portions of the bet
<code>placedDate</code>	dateTime	Date and time of bet placement
<code>price</code>	Double	Price of the remaining bet
<code>profitAndLoss</code>	Double	Net result of bet
<code>selectionId</code>	Integer	Id of the selection (this will be the same for the same selection across markets)
<code>selectionName</code>	String	Name of the selection
<code>settledDate</code>	dateTime	Date and time of bet settlement
<code>remainingSize</code>	Double	Remaining unmatched, lapsed or cancelled amount of the bet
<code>requestedSize</code>	Double	Original stake amount of the bet
<code>voidedDate</code>	dateTime	Date and time that the bet was voided (null if not applicable)
<code>executedBy</code>	String	This is used internally and always returns UNKNOWN
<code>handicap</code>	Double	The handicap selection when the <code>asianLineId</code> is > 0. You can use this to construct a unique string for the selection. I.e., "Chelsea + 3"
<code>marketTypeVariant</code>	MarketTypeVariantEnum	The Market variant for Asian Handicap markets (for possible values, see Chapter 45).

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

Table 19-4: ArrayOfMatch 0 or more instances of Match

Parameter	Type	Description
betStatus	BetStatusEnum	Status of the bet. For more information, see Chapter 45.
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 19-5: GetBetHistoryErrorEnum

Value	Condition
OK	
API_ERROR	General API Error
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.

Chapter 20 Get Bet (exchange)

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

Service Details

SOAP Action: `getBet`

Input

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

Table 20-1: 1 instance of `GetBetReq`

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
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 20-2: 1 Instance of `GetBetResp`

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
bet	Bet	The bet details
errorCode	GetBetErrorEnum	Service specific API error. See below.

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

Table 20-3: 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)
betId	Integer	Unique identifier generated for every bet placement
betStatus	BetStatusEnum	Status of the bet - please see Chapter 45.
betType	BetTypeEnum	Back or lay - please see Chapter 45.
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)

Table 20-3: 1 instance of Bet

Parameter	Type	Description
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 45.
matchedDate	dateTime	Date and time that the bet was matched (null if not applicable)
matchedSize	Double	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 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
requestedSize	Double	Original stake amount of the 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	MarketTypeVariantEnum	The Market variant for Asian Handicap markets

[Bet](#) contains an array of [Match](#). The table below shows the fields in [Match](#)

Table 20-4: ArrayOfMatch 0 or more instances of Match

Parameter	Type	Description
betStatus	BetStatusEnum	Status of the bet. For more information, see Chapter 45.
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

Table 20-4: *ArrayOfMatch* 0 or more instances of *Match*

Parameter	Type	Description
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 20-5: *GetBetErrorEnum*

Value	Condition
OK	
API_ERROR	General API Error
BET_ID_INVALID	Bet Id is invalid or does not exist. Make sure you have sent your request to the correct exchange server. The <i>getBet</i> 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.

Chapter 21

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 21-1: 1 instance of [GetMarketProfitAndLossReq](#)

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
includeSettledBets	N	Boolean	If yes, bets that have already been settled should be returned as part of P&L, default is FALSE.
marketID	Y	Integer	The market ID for which the profit and loss for the user is to be returned
netOfCommission	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 21-2: 1 Instance of [GetMarketProfitAndLossResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
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	GetMarketProfitAndLossErrorEnum	Specific error code (see enum below)

Table 21-2: 1 Instance of [GetMarketProfitAndLossResp](#)

Parameter	Type	Description
includesSettledBets	Boolean	TRUE if and only if any settled bets are included in the P&L position (regardless of the Settled Bets input)
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 45.
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 21-3: [ArrayOfProfitAndLoss](#) 0 or more instances of [ProfitAndLoss](#)

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

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 21-4: [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

Table 21-4: HandicapLine

Parameter	Type	Description
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 21-5: 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
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 21-6: GetMarketProfitAndLossErrorEnum

Value	Condition
OK	
API_ERROR	General API Error
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 only currently supported for Odds and Asian Handicap markets

Chapter 22

Get Market Traded Volume (exchange)

The API [GetMarketTradedVolume](#) service allows you to obtain all the current odds and matched amounts on the runners in a particular event.

Service Details

SOAP Action: [getMarketTradedVolume](#)

Input

The following table describes the parameters used for calling the [GetMarketTradedVolume](#) service.

Table 22-1: 1 instance of [GetMarketTradedVolumeReq](#)

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
asianLineId	N	Integer	Mandatory if the market specified by Market ID is an Asian Market, otherwise optional
currencyCode	Y	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 22-2: 1 Instance of [GetMarketTradedVolumeResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
errorCode	GetMarketTradedVolumeErrorEnum	Specific error code (see enum below)
minorErrorCode	String	Reserved for future use - currently always null
priceItems	ArrayOfVolumeInfo	Odds and matched totals data.

The [GetMarketTradedVolume](#) service returns [priceItems](#) as an array of [VolumeInfo](#). The table below shows the fields in [VolumeInfo](#)

Table 22-3: [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

Error Codes

The following table describes the error codes returned by [GetMarketTradedVolume](#).

Table 22-4: GetMarketTradedVolumeErrorEnum

Value	Condition
OK	
API_ERROR	General API Error
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

Chapter 23 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 23-1: 1 instance of `GetCurrenciesReq`

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).

Output

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

Table 23-2: 1 Instance of `GetCurrenciesResp`

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
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 23-3: `ArrayOfCurrency` 0 or more instances of `Currency`

Parameter	Type	Description
currencyCode	String	Three letter ISO 4217 code
rateGBP	Double	Exchange rate in GBP for the above currency

Chapter 24

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 24-1: 1 instance of `ConvertCurrencyReq`

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
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 24-2: 1 Instance of `ConvertCurrencyResp`

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
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 24-3: `ConvertCurrencyErrorEnum`

Value	Condition
OK	
API_ERROR	General API Error
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

Part IV

Bet Placement API Services Reference

Chapter 25

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 Chapter 44). Also, your wallet must be active (and not, for example, suspended for any reason) at the time you place the bet.

Service Details

SOAP Action: `placeBets`

Input

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

Table 25-1: 1 instance of `PlaceBetsReq`

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
bets	Y	<code>ArrayOfPlaceBets</code>	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 25-2: `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.
betType	BetTypeEnum	Bet type (back, lay or equivalent Asian Handicap types. - please see Chapter 45).
marketId	Integer	The ID of the market to place the bets on. Set to 0 for Asian Handicap Markets.
price	Double	The odds you want to set for the bet
selectionId	Integer	ID of desired Selection within the market
size	Double	The amount of the bet

Output

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

Table 25-3: 1 Instance of `PlaceBetsResp`

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
betResults	ArrayOfBetPlacementResult	Information about individual bet placement results (or errors)
errorCode	PlaceBetsErrorEnum	For errors concerning more than one bet placement request data. Errors for an individual bet are contained in the individual BetPlacementResult
minorErrorCode	String	Reserved for future use - currently always null

The **PlaceBets** service returns `betResults` as an array of [BetPlacementResult](#). The table below shows the fields in [BetPlacementResult](#)

Table 25-4: ArrayOfBetPlacementResult 1 to 60 instances of `BetPlacementResult`

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 edit.
sizeMatched	Double	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 25-5: `PlaceBetsErrorEnum`

Value	Condition
OK	
ACCOUNT_CLOSED	Account is closed - please contact BDP support
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
AUTHORISATION_PENDING	Account is pending authorisation. If the <code>PlaceBetsResultEnum</code> is also <code>CANNOT_ACCEPT_BET</code> , 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

Table 25-5: PlaceBetsErrorEnum

Value	Condition
BETWEEN_1_AND_60_BETS_REQUIRED	Number of BetPlacement less than 1 or greater than 60
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
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 25-6: PlaceBetsResultEnum

Value	Condition
OK	
ACCOUNT_CLOSED	Account is closed - please contact BDP support
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.
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.
EXPOSURE_CALCULATION_IN_PROGRESS	Exposure calculation already in progress. Bet not placed
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	Price not in betting increment
INVALID_INCREMENTS	The Price is not in allowable increments
INVALID_MARKET	MarketID does not exist

Table 25-6: PlaceBetsResultEnum

Value	Condition
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
LINES_OUT_OF_RANGE	The lines are in the wrong range and increments
LOSS_LIMIT_EXCEEDED	The potential loss for the updated bet exceeds the loss limit for the account
SELECTION_REMOVED	The runner has been removed from the event
UNKNOWN_ERROR	An unknown error occurred
VACANT_TRAP	The selected trap is vacant

Note: If Success is true, then bet has been placed. Further information may be available in errorCode.
 If BetPlacementResp.errorCode is not null, then all bet placements failed and betResults is null.
 If a single BetPlacementResult.errorCode is not null, then all bet placements failed. If success is true, then the number of betResults equals the number of bet requests.

Warning: Betting in-play is not allowed from Australia
 Betting on events other than horse racing from Holland is not allowed.

Chapter 26

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 44). Also, your wallet must be active (and not, for example, suspended for any reason) at the time you send the `UpdateBets` request.

Warning: 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 `newPrice = 200`, `newSize = 2`. The original bet is cancelled and a new bet is place for £100 with odds of 2.

Service Details

SOAP Action: `updateBets`

Input

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

Table 26-1: 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 Chapter 5).
<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 26-2: `ArrayOfUpdateBets` 1 to 15 instances of `UpdateBets`

Parameter	Type	Description
<code>betId</code>	Long	The unique identifier for the bet
<code>newPrice</code>	Double	New odds desired on the bet
<code>newSize</code>	Double	New stake desired on the bet
<code>oldPrice</code>	Double	Original odds on the bet
<code>oldSize</code>	Double	Original stake on the bet

Output

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

Table 26-3: 1 Instance of `UpdateBetsResp`

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
betResults	ArrayOfUpdateBetsResult	Details of results. See below.
errorCode	UpdateBetsErrorEnum	Specific error code (see below)
minorErrorCode	String	Reserved for future use - currently always null

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

Table 26-4: ArrayOfUpdateBetsResult 1 to 15 instances of `UpdateBetsResult`

Parameter	Type	Description
betId	Integer	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	UpdateBetsResultEnum	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 26-5: `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 updateBet 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.

Table 26-5: UpdateBetsErrorEnum

Value	Condition
API_ERROR	General API Error
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. Currently, this includes Line and Range markets.

Table 26-6: UpdateBetsResultEnum

Value	Condition
OK	
BET_CANCELLED_REMAINING_UNMATCHED	Bet partially cancelled, remainder is unmatched
BET_CANNOT_BE_ACCEPTED	Bet size cannot be accepted
BET_NOT_CANCELLED	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 updateBet 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
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	Your exposure is already being calculated, please try again later
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

Table 26-6: UpdateBetsResultEnum

Value	Condition
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.
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
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.

Warning: Betting on in-play event from Australia is forbidden.
Betting from Holland on events other than horse racing is forbidden.

Chapter 27

Cancel Bets (exchange)

The API `CancelBets` service allows you to cancel multiple (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, `cancelBet` 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 27-1: 1 instance of `CancelBetsReq`

Parameter	Mandatory?	Type	Description
<code>header</code>	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
<code>bets</code>	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 27-2: `ArrayOfCancelBets` 1 to 40 instances of `CancelBets`

Parameter	Type	Description
<code>betId</code>	<code>Long</code>	Unique identifier of bet to be cancelled

Output

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

Table 27-3: 1 Instance of `CancelBetsResp`

Parameter	Type	Description
<code>header</code>	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
<code>betResults</code>	<code>ArrayOfCancelBetsResult</code>	Between 1 and 40 <code>CancelBetsResults</code> items
<code>errorCode</code>	CancelBetsErrorEnum	Specific error code (see below)
<code>minorErrorCode</code>	<code>String</code>	Reserved for future use - currently always null

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

Table 27-4: ArrayOfCancelBetsResult 1 to 40 instances of CancelBetsResult

Parameter	Type	Description
betId	Long	The unique bet identifier
resultCode	CancelBetsResultEnum	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 27-5: CancelBetsErrorEnum

Value	Condition
OK	
API_ERROR	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.
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 27-6: CancelBetsResultEnum

Value	Condition
OK	
BET_NOT_CANCELLED	Internal 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

Table 27-6: CancelBetsResultEnum

Value	Condition
INVALID_BET_ID	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 cannot return information about bets placed on the Australian exchange server, and vice versa.
REMAINING_CANCELLED	Bet partially matched
SITE_UPGRADE	The site is not available while it is being upgraded
TAKEN_OR_LAPSED	Bet already matched or lapsed
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.

Part V

Account Management API Services Reference

Chapter 28 Create Account (global)

The API `CreateAccount` service allows you to create Betfair accounts for users. This service is restricted to certain Betfair partners.

Service Details

SOAP Action: `createAccount`

Input

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

Table 28-1: 1 instance of `CreateAccountReq`

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
acceptedPrivacyPolicyVersion	Y	Integer	The version of the privacy policy that the customer has accepted.
acceptedTermsAndConditionsVersion	Y	Integer	The version of the terms and conditions that the customer has accepted
accountType	Y	AccountTypeEnum	The type of account to create
address1	Y	String	The first line of the customer's address
address2	N	String	The second line of the customer's address
address3	N	String	The third line of the customer's address
answer1	Y	String	The answer to the first security question
answer2	Y	String	The answer to the second security question
browser	N	String	The customer's browser identification string, if applicable
countryOfResidence	Y	String	The customer's country in ISO3 format.

Table 28-1: 1 instance of [CreateAccountReq](#)

Parameter	Mandatory?	Type	Description
countyState	Y	String	The customer's county or state. Note that, for customers whose countryOfResidence is Australia, this 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.
currency	Y	String	The account currency
dateOfBirth	Y	DateTime	The customer's birth date
depositLimit	N	Double	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.)
emailAddress	Y	String	The customer's e-mail address
firstName	Y	String	The customer's first name
gender	Y	GenderEnum	The customer's gender
homeTelephone	Y	String	The customer's home telephone number
informProductsServices	Y	Boolean	If the customer wants to be contacted about products and services
informSpecialOffers	Y	Boolean	If the customer wants to be contacted about special offers
ipAddress	Y	String	The end-user customer's IP address.

Table 28-1: 1 instance of [CreateAccountReq](#)

Parameter	Mandatory?	Type	Description
locale	Y	String	The customer's locale
lossLimit	N	Double	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.)
manualAddress	Y	Boolean	Set to true if the address was entered manually. Set to false if you have validated the address in some way
mobileTelephone	N	String	The customer's mobile telephone number
partnerID	Y	Integer	The partner ID
password	Y	String	The desired password
postCode	Y	String	The customer's postal code
preferredName	N	String	The name by which the customer prefers to be addressed in correspondence with Betfair.
productID	Y	String	The product ID
question1	Y	SecurityQuestion1Enum	The first security question used
question2	Y	SecurityQuestion2Enum	The second security question used
referrerCode	N	String	A refer and earn or promotion code
region	N	RegionEnum	The customer's region
subPartnerID	N	Integer	The sub-partner ID
superPartnerID	N	Integer	The super-partner ID
surname	Y	String	The customer's surname

Table 28-1: 1 instance of [CreateAccountReq](#)

Parameter	Mandatory?	Type	Description
timeZone	Y	String	The customer's time zone
title	Y	TitleEnum	The customer's title
townCity	Y	String	The customer's town or city
username	Y	String	The desired username for the account
workTelephone	N	String	The customer's work telephone

Output

The following table describes the parameters returned from the [createAccount](#) service.

Table 28-2: 1 Instance of [CreateAccountResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
accountID	Integer	The new account ID
accountStatus	AccountStatusEnum	The current status of the account
errorCode	CreateAccountErrorEnum	Service specific error code. See below.
minorErrorCode	String	Reserved for future use. Currently always nil.
tan	String	The Telbet account code
userId	Integer	The new account user ID
validationErrors	ValidationErrorsEnum	An array containing validation errors for the submitted profile modifications

Error Codes

The following table describes the error codes returned by [createAccount](#).

Table 28-3: [CreateAccountErrorEnum](#)

Value	Condition
OK	
ACCOUNT_CREATION_ERROR	There was an unknown error when creating the account
API_ERROR	General API error

Table 28-3: CreateAccountErrorEnum

Value	Condition
VALIDATION_ERRORS	There were problems with the submitted information. Check the validationErrors array for specific problems. Note that, for users whose countryOfResidence is Australia, the “countyState” input parameter must specify the state (not the county). The valid values for it are: Australian Capital Territory, New South Wales, Victoria, Queensland, South Australia, Western Australia, Tasmania. If one of these values is not entered for a request, a VALIDATION_ERRORS message will be returned.

Chapter 29 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 `submitLIMBMessage` 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 29-1: 1 instance of `RetrieveLIMBMessageReq`

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).

Output

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

Table 29-2: 1 Instance of `RetrieveLIMBMessageResp`

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
errorCode	RetrieveLIMBMessageErrorEnum	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	RetrievePersonalMessage	Personal messages
retrieveTCPrivacyPolicyChangeMessage	RetrieveTCPrivacyPolicyChangeMessage	Policy change messages
retrievePasswordChangeMessage	RetrievePasswordChangeMessage	Password change messages
retrieveBirthDateCheckMessage	RetrieveBirthDateCheckMessage	Birthdate confirmation
retrieveAddressCheckMessage	RetrieveAddressCheckMessage	Address confirmation

Table 29-2: 1 Instance of [RetrieveLIMBMessageResp](#)

Parameter	Type	Description
retrieveContactDetailsCheckMessage	RetrieveContactDetailsCheckMessage	Contact details confirmation
retrieveChatNameChangeMessage	RetrieveChatNameChangeMessage	Chat name change request
retrieveCardBillingAddressCheckItems	ArrayOfRetrieveCardBillingAddressCheckLIMBMessage	Card address confirmation

The [retrieveLIMBMessage](#) service returns a number of message blocks. The tables below shows the fields for each message type

Table 29-3: 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 29-4: 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 29-5: 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 29-6: 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 29-7: 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 29-8: 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 29-9: 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 29-10: ArrayOfRetrieveCardBillingAddressCheckLIMBMessage

Parameter	Type	Description
messageID	Int	The ID of the message

Table 29-10: ArrayOfRetrieveCardBillingAddressCheckLIMBMessage

Parameter	Type	Description
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
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 29-11: RetrieveLIMBMessageErrorEnum

Value	Condition
OK	
API_ERROR	General API error

Chapter 30 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 30-1: 1 instance of [SubmitLIMBMessageReq](#)

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
password	Y	String	The customer's password
submitPersonalMessage	N	SubmitPersonalMessage	Personal message response
submitTCPrivacyPolicyChangeMessage	N	SubmitTCPrivacyPolicyChangeMessage	Policy change response
submitPasswordChangeMessage	N	SubmitPasswordChangeMessage	Password change
submitBirthDateCheckMessage	N	SubmitBirthDateCheckMessage	Birth date confirmation
submitAddressCheckMessage	N	SubmitAddressCheckMessage	Address confirmation
submitContactDetailsCheckMessage	N	SubmitContactDetailsCheckMessage	Contact details confirmation
submitChatNameChangeMessage	N	SubmitChatNameChangeMessage	Chat name change
submitCardBillingAddressCheckItems	N	ArrayOfSubmitCardBillingAddressCheckLIMBMessage	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 30-2: [SubmitPersonalMessage](#)

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

Table 30-3: SubmitTCPrivacyPolicyChangeMessage

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

Table 30-4: 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 30-5: 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 30-6: 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 30-7: 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).

Table 30-7: SubmitContactDetailsCheckMessage

Parameter	Type	Description
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 30-8: 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 30-9: 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 30-10: 1 Instance of SubmitLIMBMessageResp

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
errorCode	SubmitLIMBMessageErrorEnum	Service specific error code. See below

Table 30-10: 1 Instance of `SubmitLIMBMessageResp`

Parameter	Type	Description
minorErrorCode	String	Reserved for future use. Currently always null.
validationErrors	LIMBValidationErrorsEnum	An array containing validation errors for the submitted message responses

Error Codes

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

Table 30-11: `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

The `submitLIMBMessage` service returns [LIMBValidationErrorsEnum](#) as an array. The table below shows the fields in the array of [LIMBValidationErrorsEnum](#).

Table 30-12: `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.
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

Table 30-12: LIMBValidationErrorsEnum

Value	Condition
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
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.

Table 30-12: LIMBValidationErrorsEnum

Value	Condition
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.

Chapter 31 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 31-1: 1 instance of [ViewProfileReq](#)

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).

Output

The following table describes the parameters returned from the [viewProfile](#) service.

Table 31-2: 1 Instance of [ViewProfileResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
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

Table 31-2: 1 Instance of `ViewProfileResp`

Parameter	Type	Description
<code>mobileTelephone</code>	String	The mobile telephone number of the account holder
<code>emailAddress</code>	String	The registered e-mail address for the account
<code>timeZone</code>	String	The account holder's time zone
<code>currency</code>	String	The currency used for the account
<code>gamcareLimit</code>	Integer	The Gamcare deposit limit for the account. (This parameter corresponds to the <code>depositLimit</code> parameter in the <code>CreateAccount</code> and <code>ModifyProfile</code> services.)
<code>gamcareFrequency</code>		Whether the Gamcare deposit limit is so much per day, per week, per month, or per year. (This parameter corresponds to the <code>depositLimitFrequency</code> parameter in the <code>CreateAccount</code> and <code>ModifyProfile</code> services.)
<code>gamcareLossLimit</code>	Integer	The Gamcare loss limit for the account. (This parameter corresponds to the <code>lossLimit</code> parameter in the <code>CreateAccount</code> and <code>ModifyProfile</code> services.)
<code>gamcareLossLimitFrequency</code>		Whether the Gamcare loss limit is so much per day, per week, per month, or per year. (This parameter corresponds to the <code>lossLimitFrequency</code> parameter in the <code>CreateAccount</code> and <code>ModifyProfile</code> services.)
<code>gamcareUpdateDate</code>	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 31-3: `ViewProfileErrorEnum`

Value	Condition
OK	
API_ERROR	General API error
UNAUTHORIZED	The current user is not permitted to view the account profile

Chapter 32 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 32-1: 1 instance of `ModfyProfileReq`

Parameter	Mandatory?	Type	Description
<code>header</code>	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
<code>password</code>	Y	String	The account password
<code>address1</code>	N	String	The first line of the account address
<code>address2</code>	N	String	The second line of the account address
<code>address3</code>	N	String	The third line of the account address
<code>townCity</code>	N	String	The town or city of the account address
<code>countyState</code>	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.
<code>postCode</code>	N	String	The postal code of the account address
<code>countryOfResidence</code>	N	String	The country of the account address
<code>homeTelephone</code>	N	String	The home telephone number of the account holder
<code>workTelephone</code>	N	String	The work telephone number of the account holder
<code>mobileTelephone</code>	N	String	The mobile telephone number of the account holder
<code>emailAddress</code>	N	String	The registered e-mail address for the account
<code>timeZone</code>	N	String	The account holder's time zone
<code>depositLimit</code>	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.)

Table 32-1: 1 instance of [ModifyProfileReq](#)

Parameter	Mandatory?	Type	Description
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.)

Output

The following table describes the parameters returned from the [modifyProfile](#) service.

Table 32-2: 1 Instance of [ModifyProfileResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
errorCode	ModifyProfileErrorEnum	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 32-3: [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 customers, check that the countyState parameter has been correctly specified in the ModifyProfile request (see Table 32-1 on page 103).
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

Chapter 33 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 13.

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 33-1: 1 instance of `AccountFundsReq`

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).

Output

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

Table 33-2: 1 Instance of `AccountFundsResp`

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
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

Table 33-2: 1 Instance of `AccountFundsResp`

Parameter	Type	Description
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 populated for the Australian wallet.)
withdrawBalance	Decimal	Balance available for withdrawal
errorCode	GetAccountFundsErrorEnum	Specific error code (see Table 33-3 on page 106)

Error Codes

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

Table 33-3: GetAccountFundsErrorEnum

Value	Condition
OK	
EXPOSURE_CALC_IN_PROGRESS	The API is busy calculating your current exposure.
API_ERROR	General API error.

Chapter 34

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 34-1: 1 instance of `ModifyPasswordReq`

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
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 34-2: 1 Instance of `ModifyPasswordResp`

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
errorCode	ModifyPasswordErrorEnum	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 34-3: `ModifyPasswordErrorEnum`

Value	Condition
OK	
API_ERROR	General API error.
INVALID_NEW_PASSWORD	The new password supplied is not valid
INVALID_PASSWORD	The existing password supplied is not correct

Table 34-3: ModifyPasswordErrorEnum

Value	Condition
PASSWORDS_DONT_MATCH	The supplied new passwords do not match each other.

Chapter 35 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 35-1: 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 Chapter 5).
<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

Table 35-1: 1 instance of [ForgotPasswordReq](#)

Parameter	Mandatory?	Type	Description
forgottenPasswordAnswer2	N	String	The answer to the second security question
newPassword	N	String	The new password for the account
newPasswordRepeat	N	String	The new password for the account

Output

The following table describes the parameters returned from the [forgotPassword](#) service.

Table 35-2: 1 Instance of [ForgotPasswordResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
errorCode	ForgotPasswordErrorEnum	Service specific error code. See below.
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 35-3: [ForgotPasswordErrorEnum](#)

Value	Condition
OK	
API_ERROR	General API Error
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

Chapter 36 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 described in .) 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.

Input

The following table describes the parameters used for calling the [GetAccountStatement](#) service.

Table 36-1: 1 instance of [GetAccountStatementReq](#)

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
startRecord	Y	Integer	The first record number to return (supports paging)
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 45.
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 36-2: 1 Instance of [GetAccountStatementResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
items	ArrayOfAccountStatementItem	Market type (see below)
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 36-3: [ArrayOfAccountStatementItem](#) 0 or more instances of [AccountStatementItem](#)

Parameter	Type	Description
accountBalance	BigDecimal	Account Balance
Amount	BigDecimal	The amount won / lost for bets or amount deposited / withdrawn in the account currency
avgPrice	BigDecimal	The average matched price of the bet (null if no part has been matched)
betId	Integer	Unique identifier generated for every bet placement
betSize	BigDecimal	The amount of the stake of your bet. (0 for commission payments or deposit/withdrawals)
betType	BetTypeEnum	Back or lay - please see Chapter 45 for details.
commissionRate	String	Commission rate on market
eventId	Integer	Id of the market
eventTypeid	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 45. For account deposits and withdrawals, 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

Table 36-3: `ArrayOfAccountStatementItem` 0 or more instances of `AccountStatementItem`

Parameter	Type	Description
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 45 for details.
winLose	AccountStatementEnum	Win or loss - please see Chapter 45 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 36-4: `GetAccountStatementErrorEnum`

Value	Condition
OK	
API_ERROR	General API Error
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

Chapter 37

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 37-1: 1 instance of [GetSubscriptionInfoReq](#)

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).

Output

The following table describes the parameters returned from the [GetSubscriptionInfo](#) service.

Table 37-2: 1 Instance of [GetSubscriptionInfoResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
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 37-3: [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 45.
productId	Integer	Product ID
productName	String	The name of the subscription product
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 45.
subscribedDate	dateTime	The date the subscription was enabled

Table 37-3: ArrayOfSubscription 0 or more instances of Subscription

Parameter	Type	Description
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 37-4: 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 45 .
usageCount	Integer	Current usage count

Chapter 38

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 38-1: 1 instance of [addPaymentCardReq](#)

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
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 45). 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.
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 38-2: 1 Instance of [addPaymentCardResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
errorCode	AddPaymentCardErrorEnum	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 38-3: 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
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

Table 38-3: 1 instance of `PaymentCard`

Parameter	Type	Description
<code>billingCountryIso3</code>	String	The ISO3 country code for the card billing address
<code>cardStatus</code>		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 `addPaymentCard`.

Table 38-4: `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	There was a general API error. Check the API header.
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
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 45-9 on page 137.
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

Table 38-4: AddPaymentCardErrorEnum

Value	Condition
UNAUTHORIZED	The account is not authorised to add a payment card

Chapter 39

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 39-1: 1 instance of `GetPaymentCardReq`

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).

Output

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

Table 39-2: 1 Instance of `GetPaymentCardResp`

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
errorCode	GetPaymentCardErrorEnum	Service specific error message. See below.
minorErrorCode	String	Reserved for future use - currently always nulls
paymentCardItems	ArrayOfPaymentCard	An array of type PaymentCard containing details of registered cards.

The `getPaymentCard` service returns [PaymentCard](#). The table below shows the fields in [PaymentCard](#)

Table 39-3: 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.)
issuingCountryIso3	String	The country where the card was issued

Table 39-3: 1 instance of [PaymentCard](#)

Parameter	Type	Description
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 39-4: [GetPaymentCardErrorEnum](#)

Value	Condition
OK	
ACCOUNT_INACTIVE	The account is suspended or locked
API_ERROR	This is a general API error
INVALID_PASSWORD	The password is not valid
UNAUTHORIZED	The account is not authorised to manage payment cards

Chapter 40

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 40-1: 1 instance of `DeletePaymentCardReq`

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
nickName	Y	String	An arbitrary name used for referring to this card
password	Y	String	The Betfair account login password

Output

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

Table 40-2: 1 Instance of `DeletePaymentCardResp`

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
errorCode	DeletePaymentCardErrorEnum	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 MMY

Error Codes

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

Table 40-3: `DeletePaymentCardErrorEnum`

Value	Condition
OK	

Table 40-3: DeletePaymentCardErrorEnum

Value	Condition
ACCOUNT_INACTIVE	The account is locked or suspended
API_ERROR	A general API error occurred
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

Chapter 41

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 41-1: 1 instance of `UpdatePaymentCardReq`

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
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).
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	PaymentCardStatus Enum	The card's status. It can be either LOCKED or UNLOCKED (see Chapter 45). 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 <code>GetPaymentCard</code> service (see Chapter 39).

Output

The following table describes the parameters returned from the [updatePaymentCard](#) service.

Table 41-2: 1 Instance of [UpdatePaymentCardResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
errorCode	UpdatePaymentCardErrorEnum	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 41-3: [UpdatePaymentCardErrorEnum](#)

Value	Condition
OK	
ACCOUNT_INACTIVE	The Betfair account is suspended or locked
API_ERROR	This is a general API error
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

Table 41-3: UpdatePaymentCardErrorEnum

Value	Condition
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.)

Chapter 42 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 44).

Service Details

SOAP Action: `depositFromPaymentCard`

Input

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

Table 42-1: 1 instance of `DepositFromPaymentCardReq`

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
amount	Y	Double	The amount to deposit from the payment card
cardIdentifier	Y	String	The nickname for the payment card
cv2	Y	String	The CV2 digits from the card
password	Y	String	The account password

Output

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

Table 42-2: 1 Instance of `DepositFromPaymentCardResp`

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
errorCode	PaymentsErrorEnum	Service specific error codes. See below.
fee	Double	The card transaction fee for this deposit
maxAmount	Double	The maximum deposit amount allowed for this card. If <code>errorCode</code> is set to <code>CARD_AMOUNT_OUTSIDE_LIMIT</code> then <code>minAmount</code> and <code>maxAmount</code> is set. If <code>errorCode</code> is set to <code>DEPOSIT_LIMIT_EXCEEDED</code> then <code>maxAmount</code> is set.
minAmount	Double	The minimum deposit amount allowed for this card. If <code>errorCode</code> is set to <code>CARD_AMOUNT_OUTSIDE_LIMIT</code> then <code>minAmount</code> and <code>maxAmount</code> is set. If <code>errorCode</code> is set to <code>DEPOSIT_LIMIT_EXCEEDED</code> then <code>maxAmount</code> is set.
minorErrorCode	String	Not used
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 42-3: PaymentsErrorEnum

Value	Condition
OK	
ACCOUNT_SUSPENDED	The Betfair account is currently suspended
API_ERROR	General API Error
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
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.

Chapter 43

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 43-1: 1 instance of [WithdrawToPaymentCardReq](#)

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
amount	Y	Double	The amount of funds to withdraw
cardIdentifier	Y	String	The nickname for the card
password	Y	String	The account password

Output

The following table describes the parameters returned from the [withdrawToPaymentCard](#) service.

Table 43-2: 1 Instance of [WithdrawToPaymentCardResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
amountWithdrawn	Double	The amount successfully withdrawn to the payment card
errorCode	PaymentsErrorEnum	Service specific error message. See below.
maxAmount	Double	The maximum amount you can withdraw to this payment card in a single transaction.
minorErrorCode	String	Reserved for future use - currently always null

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 43-3: [PaymentsErrorEnum](#)

Value	Condition
OK	
ACCOUNT_SUSPENDED	The Betfair account is currently suspended

Table 43-3: PaymentsErrorEnum

Value	Condition
API_ERROR	General API Error
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
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.

Chapter 44 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 "The Architecture of API 5.x" on page 11.

Service Details

SOAP Action: [transferFunds](#)

Input

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

Table 44-1: 1 instance of [TransferFundsReq](#)

Parameter	Mandatory?	Type	Description
header	Y	APIRequestHeader	The API Request Header that contains the user's session token (see Chapter 5).
sourceWalletId	Y	INT	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	INT	The wallet that you are requesting the funds to be transferred into. There are two possible wallets: 1 = UK Sports Betting wallet 2 = Australian Sports Betting wallet
amount	Y	Double	The amount of money you want to transfer.

Output

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

Table 44-2: 1 Instance of [TransferFundsResp](#)

Parameter	Type	Description
header	APIResponseHeader	The API Response Header that contains the updated session token (see Chapter 5).
depositsThisMonth	Double	The total value of all deposits into the specified wallet so far in the current month. If the transfer request that occasioned the response you received was successful, then the value of <code>depositsThisMonth</code> includes the sum that you specified in your transfer request.
errorCode	TransferFundsErrorEnum	Service-specific error message. See below.

Error Codes

The following table describes the error codes returned by [TransferFunds](#).

Table 44-3: TransferFundsErrorEnum

Value	Condition
OK	The transfer you requested was successful.
SOURCE_WALLET_DOES_NOT_EXIST	The value that you specified for the source wallet was invalid. You must specify 1 for the UK wallet, and 2 for the Australian wallet.
DESTINATION_WALLET_DOES_NOT_EXIST	The value that you specified for the destination wallet was invalid. You must specify 1 for the UK wallet, and 2 for the Australian wallet.
INSUFFICIENT_FUNDS	There are insufficient funds in your source wallet for the requested transfer to be performed.
DEPOSIT_LIMIT_NONKYC	<p>During an initial setup period - which can last for a maximum of three months after the first time you transfer funds into your Australian wallet - there is a limit of \$300 (USD) per month on the amount you can transfer into your Australian wallet.</p> <p>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 \$300 (USD) per month deposit limit is lifted.</p> <p>If you receive this error message it means that you are attempting to transfer more than \$300 (USD) into your Australian wallet while the limit is still in force.</p>
CUMULATIVE_DEPOSIT_NONKYC	You are attempting to transfer a sum of money into your Australian wallet which, when added to previous transfers you have made in the current month, exceeds the monthly limit of \$100 (USD) that you are currently permitted to transfer into your Australian wallet (see above). This limit will be lifted when Betfair have received and processed the personal information that they have requested from you.
SOURCE_DESTINATION_IDENTICAL	You have specified the same wallet as both the source and the destination for the transfer.
KYC_WITHDRAWALS_BLOCKED	Until Betfair have received and processed the personal information they have requested from you, Australian law forbids you to withdraw any funds from your Australian wallet.
SOURCE_WALLET_BLOCKED	A block has been placed on withdrawals from this wallet. Please contact Betfair.
DESINTATION_WALLET_BLOCKED	A block has been placed on deposits into this wallet. Please contact Betfair.

Part VI

Appendix

Chapter 45 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 45-1: 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 45-2: 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 45-3: AccountStatusEnum

Value	Description
A	Account active

Table 45-3: AccountStatusEnum

Value	Description
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 45-4: 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

BetsOrderByEnum

The following table describes the BetsOrderByEnum simple data type.

Table 45-5: 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 45-6: BetStatusEnum

Value	Description
C	Cancelled
L	Lapsed
M	Matched
MU	Matched and Unmatched
S	Settled
U	Unmatched
V	Voided

BetTypeEnum

The following table describes the BetTypeEnum simple data type.

Table 45-7: BetTypeEnum

Value	Description
B	Back
L	Lay

BillingPeriodEnum

The following table describes the BillingPeriodEnum simple data type.

Table 45-8: 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 45-9: 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 45-10: 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 45-11: GenderEnum

Value	Description
F	Female
M	Male

MarketStatusEnum

The following table describes the MarketStatusEnum simple data type.

Table 45-12: 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 45-13: 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 45-14: MarketTypeVariantEnum

Value	Description
D	Default
ASL	Asian Single Line
ADL	Asian Double Line

PaymentCardStatusEnum

The following table describes the PaymentCardStatusEnum simple data type.

Table 45-15: 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 45-16: 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 45-17: 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 45-18: 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 45-19: 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

Table 45-19: ServiceEnum

Value	Description
LOAD_MARKET_PRICES	Access to getMarketPrices service
LOAD_MARKET_PRICES_COMPRESSED	Access to getMarketPricesCompressed service
LOAD_MARKET_PROFIT_LOSS	Access to getMarketProfitAndLoss service
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 45-20: SortOrderEnum

Value	Description
ASC	Sort ascending
DESC	Sort descending

SubscriptionStatusEnum

The following table describes the SubscriptionStatusEnum simple data type.

Table 45-21: SubscriptionStatusEnum

Value	Description
ACTIVE	Subscription Active
INACTIVE	Subscription Inactive
SUSPENDED	Subscription Suspended

TitleEnum

The following table describes the TitleEnum simple data type.

Table 45-22: TitleEnum

Value	Description
DR	Dr.
MISS	Miss.
MR	Mr.
MRS	Mrs.
MS	Ms.

ValidationErrorsEnum

The following table describes the ValidationErrorsEnum simple data type.

Table 45-23: 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

Table 45-23: ValidationErrorsEnum

Value	Condition
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
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 black 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

Chapter 46 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.

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.

Line and Range Markets

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

Timestamps

The timestamp in the header is the time the header was generated on the server. The timestamp on an individual object is the time the data in the object was created. Therefore for SportItems, this could be up to 20 minutes old. For Market, it could be up to 2 seconds old. Timestamps are accurate to 1 second.

Betfair has an internal NTP server from which all customer-facing servers are synchronized. This NTP server in turn is synchronized with the following public NTP servers:

- Chimera.csx.cam.ac.uk
- Ntp.maths.tcd.ie
- Dire.bris.ac.uk
- Ntp1.pipex.net

To compare Betfair Timestamps with your own time, you should synchronize with one of these servers.

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 $3 - ((3-1)*40\%) = 2.2$. 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 retrieved using the API via the 'InPlayDelay' field in the response to the MarketRefresh call. When the market is in-running, the InPlayDelay will be set to greater than 0.

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 (InPlayDelay, in seconds) 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 46-1: 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 47: Locale String Examples

Locale String	Locale
en	English language
en_GB	English language with United Kingdom region
haw	Hawaiian language
en_Variant	NOT VALID

Part VII

Index

Chapter 47 Index

A

- AccountFundsReq 105
- AccountFundsResp 105
- AccountStatementEnum 134
- AccountStatementIncludeEnum 134
- AccountStatementItem 112
- AccountStatusEnum 134
- AccountTypeEnum 135
- addPaymentCard 116
- APIReponseHeader 21
- APIRequestHeader 21
- arrayOfCouponLinks 42
- ArrayOfEventType 30
- Australia 77, 81
- Australian exchange server 11
- AvailabilityInfo 51

B

- bestPricesToBack 45
- bestPricesToLay 45
- Bet 53, 61, 63
- bet 63
- betHistoryItems 61
- BetPlacementResult 75
- betResults 75, 79, 83
- bets 53, 57, 74, 78
- BetsOrderByEnum 135
- BetStatusEnum 136
- Betting In-Running 145
- BetTypeEnum 136
- BillingPeriodSOAPEnum 136

C

- CancelBets 82
- cancelBets 82
- CancelBetsReq 82
- CancelBetsResp 82
- CancelBetsResult 83
- CardTypeEnum 137
- convertCurrency 72
- ConvertCurrencyReq 72
- ConvertCurrencyResp 72
- CouponLink 42
- createAccount 86
- Currency 71, 144

- currencyItems 71

D

- deletePaymentCard 122
- depositFromPaymentCard 127

E

- Event 37
- eventItems 37
- eventTypeItems 28
- exchange servers 11, 14
- exchange services 10
- exchangeId 14

F

- forgotPassword 109
- futureIfWin 66

G

- GamcareLimitFreqEnum 142
- GenderEnum 137
- getAccountFunds 105
- getAccountStatement 111
- GetAccountStatementReq 111
- GetAccountStatementResp 112
- getActiveEventTypes 28
- getAllCurrencies 71
- getAllEventTypes 30
- GetAllMarkets 32
- GetAllMarketsReq 32
- GetAllMarketsResp 32
- getBet 63
- getBetHistory 59
- GetBetHistoryReq 59
- GetBetHistoryResp 60
- GetBetReq 63
- GetBetResp 63
- GetCurrenciesReq 71
- GetCurrenciesResp 71
- getCurrentBets 52
- GetCurrentBetsReq 52
- GetCurrentBetsResp 53
- getDetailAvailableMktDepth 50

GetDetailedAvailMktDepthReq 50
 GetDetailedAvailMktDepthResp 50
 getEvents 14, 36
 GetEventsForEventTypes 39
 GetEventsReq 36
 GetEventsResp 36
 GetEventTypesReq 30
 GetEventTypesResp 30
 getMarket 14, 40
 getMarketPricesCompressed 47
 getMarketProfitAndLoss 66
 GetMarketProfitAndLossReq 66
 GetMarketProfitAndLossResp 66
 GetMarketReq 40
 GetMarketResp 40
 getMarketTradedVolume 69
 GetMarketTradedVolumeReq 69
 GetMarketTradedVolumeResp 69
 getMUBets 56
 GetMUBetsReq 56
 GetMUBetsResp 57
 getPaymentCard 120
 GetPaymentCardReq 120
 GetPaymentCardResp 120
 GetSportsForEvent 39
 getSubscriptionInfo 114
 GetSubscriptionInfoResp 114
 global services 10, 12

H

HandicapLine 67
 heartbeat 26
 Holland 77, 81
 Horse Racing Non-Runners 144

I

items 112

K

keepAlive 26
 key 18

L

Line 144
 load balancer 18
 login 23
 logout 25

M

market 41
 marketData 33
 marketItems 37
 MarketPrices 44
 marketPrices 47
 MarketStatusEnum 138
 MarketSummary 37
 MarketTypeEnum 138
 Match 54, 62, 64
 modifyPassword 107
 modifyProfile 103
 MUBet 57
 MultiWinnerOddsLine 68

N

navigation 32
 NTP 144

O

ordering 53, 57

P

PaymentCard 120
 performance 18
 PlaceBets 74
 placeBets 74
 PlaceBetsReq 74
 PlaceBetsResp 75
 Price 45
 priceItems 51, 69
 ProfitAndLoss 67

R

Range 144
 reduction 144
 RegionEnum 142
 retrieveLIMBMessage 91
 Runner 42
 runnerPrices 45
 runners 42
 RunnesPrices 45

S

SecurityQuestionIEnum 142

- SecurityQuestion2Enum 140
- ServiceCall 115
- ServiceEnum 140
- services 115
- sessionToken 18, 24
- SOAP 10
- SortOrderEnum 141
- submitLIMBMessage 95
- Subscription 114
- subscriptions 114
- SubscriptionStatusSOAPEnum 141

T

- Tasmanian Gaming Commission 11
- Timestamps 144
- Timezones 144
- TitleEnum 142
- TransferFunds 13, 15, 131

U

- UK exchange server 11
- UpdateBets 78
- updateBets 78
- UpdateBetsErrorEnum 79
- UpdateBetsReq 78
- UpdateBetsResult 79
- UpdateBetsResultEnum 80
- updatePaymentCard 124
- URLs for API services 10

V

- ValidationErrorsEnum 142
- viewProfile 101
- VolumeInfo 69

W

- wallets 13, 15, 131, 132
- withdrawToPaymentCard 129
- worstCaseIfWin 66