

# API Documentation

This documentation provides a comprehensive overview of our API services, including available endpoints and data models.

## Table of Contents

1. [REST Services](#)
2. [Real-time Services](#)
3. [Data Models](#)

## REST Services

### bonuses

#### **getGrantedBonuses**

*Retrieves the list of bonuses that have been granted to the user*

**Arguments:**

**Returns:** [GrantedBonus](#), [GrantedBonus]

#### **optOutBonus**

*Opts out from a specific bonus*

**Arguments:**

- partyId: String
- code: String

**Returns:** [BasicResponse](#)

#### **getAvailableBonuses**

*Retrieves the list of bonuses available for the user to claim*

**Arguments:**

**Returns:** [AvailableBonus](#), [AvailableBonus]

#### **redeemBonus**

*Redeems a bonus using a bonus code*

**Arguments:**

- code: String

**Returns:** [RedeemBonusResponse](#)

## **cancelBonus**

*Cancels an active bonus*

### **Arguments:**

- bonusId: String

**Returns:** [BasicResponse](#)

## **redeemAvailableBonus**

*Claims an available bonus for a specific user*

### **Arguments:**

- code: String
- partyId: String

**Returns:** [BasicResponse](#)

## **registration**

### **signUpCompletion**

*Completes the signup process with additional user information*

### **Arguments:**

- Form containing additional user information

**Returns:** Bool

### **simpleSignUp**

*Registers a new user with basic information*

### **Arguments:**

- form: SimpleSignUpForm

**Returns:** Bool

### **validateUsername**

*Validates a username and provides suggestions if unavailable*

### **Arguments:**

- username: String

**Returns:** UsernameValidation

### **checkEmailRegistered**

*Checks if an email is already registered in the system*

**Arguments:**

- email: String

**Returns:** Bool**signUp***Registers a new user with complete information***Arguments:**

- form: SignUpForm

**Returns:** SignUpResponse**signupConfirmation***Confirms user signup with verification code***Arguments:**

- confirmationCode: String
- email: String

**Returns:** Bool**profile****getUserProfile***Retrieves the user's profile information***Arguments:**

- kycExpire: String?

**Returns:** UserProfile**support****contactUs***Sends a contact request to customer support***Arguments:**

- message: String
- subject: String
- email: String
- firstName: String
- lastName: String

**Returns:** [BasicResponse](#)

## contactSupport

*Sends a detailed support request with user information*

### Arguments:

- lastName: String
- isLoggedIn: Bool
- userIdentifier: String
- firstName: String
- message: String
- subject: String
- email: String
- subjectType: String

**Returns:** [SupportResponse](#)

## location

### getAllCountries

*Retrieves all available countries*

### Arguments:

**Returns:** Country, [Country]

### getCountries

*Retrieves list of available countries*

### Arguments:

**Returns:** Country, [Country]

### getCurrentCountry

*Retrieves the current country information*

### Arguments:

**Returns:** Country?

## events

### getHighlightedLiveEvents

*Retrieves detailed information about highlighted live events, optionally filtered by user*

### Arguments:

- eventCount: Int
- userId: String?

**Returns:** [Event](#), [Event]

## **getEventSecondaryMarkets**

*Retrieves secondary markets information for a specific event*

### **Arguments:**

- eventId: String

**Returns:** [Event](#)

## **getEventDetails**

*Retrieves detailed information about a specific event*

### **Arguments:**

- eventId: String

**Returns:** [Event](#)

## **getEventsForEventGroup**

*Retrieves events associated with a specific event group*

### **Arguments:**

- withId: String

**Returns:** [EventsGroup](#)

## **getHeroGameEvent**

*Retrieves the hero game event*

### **Arguments:**

**Returns:** [Event](#), [Event]

## **getPromotionalSlidingTopEvents**

*Retrieves promotional sliding events for the top section*

### **Arguments:**

**Returns:** [Event](#), [Event]

## **getEventSummaryByMarket**

*Retrieves a summary of an event using a market ID associated with the event*

### **Arguments:**

- forMarketId: String

**Returns:** [Event](#)

## **getHighlightedMarkets**

*Retrieves highlighted markets*

**Arguments:**

**Returns:** HighlightMarket, [HighlightMarket]

## **getSearchEvents**

*Searches for events based on a query string with pagination support*

**Arguments:**

- page: String
- resultLimit: String
- query: String
- isLive: Bool

**Returns:** [EventsGroup](#)

## **getPromotionalTopBanners**

*Retrieves promotional banners for the top section*

**Arguments:**

**Returns:** [PromotionalBanner](#), [PromotionalBanner]

## **getTopCompetitions**

*Retrieves the list of top competitions*

**Arguments:**

**Returns:** TopCompetition, [TopCompetition]

## **getCashbackSuccessBanner**

*Retrieves the cashback success banner*

**Arguments:**

**Returns:** [BannerResponse](#)

## **getRegionCompetitions**

*Retrieves information about competitions available in a specific region*

**Arguments:**

- regionId: String

**Returns:** [SportRegionInfo](#)

## **getEventForMarketGroup**

*Retrieves an event associated with a specific market group*

**Arguments:**

- withId: String

**Returns:** [Event](#)

**getPromotionalTopStories**

*Retrieves promotional top stories*

**Arguments:**

**Returns:** [PromotionalStory](#), [PromotionalStory]

**getAvailableSportTypes**

*Retrieves a list of available sport types within an optional date range*

**Arguments:**

- endDate: Date?
- initialDate: Date?

**Returns:** [SportType](#), [SportType]

**getHighlightedLiveEventsIds**

*Retrieves IDs of highlighted live events, optionally filtered by user*

**Arguments:**

- userId: String?
- eventCount: Int

**Returns:** String, [String]

**getHomeSliders**

*Retrieves the home page slider banners*

**Arguments:**

**Returns:** [BannerResponse](#)

**getCompetitionMarketGroups**

*Retrieves information about market groups available for a specific competition*

**Arguments:**

- competitionId: String

**Returns:** [SportCompetitionInfo](#)

**getEventLiveData**

*Retrieves live data and statistics for a specific event*

**Arguments:**

- eventId: String

**Returns:** EventLiveData

**getPromotedSports**

*Retrieves the list of promoted sports*

**Arguments:**

**Returns:** [PromotedSport](#), [PromotedSport]

**getSportRegions**

*Retrieves information about regions available for a specific sport*

**Arguments:**

- sportId: String

**Returns:** [SportNodeInfo](#)

**getHighlightedBoostedEvents**

*Retrieves events with boosted odds that are highlighted*

**Arguments:**

**Returns:** [Event](#), [Event]

**getTopCompetitionsPointers**

*Retrieves pointers to top competitions*

**Arguments:**

**Returns:** [TopCompetitionPointer](#), [TopCompetitionPointer]

**getEventSummary**

*Retrieves a summary of a specific event using its event ID*

**Arguments:**

- eventId: String

**Returns:** [Event](#)

**getMarketInfo**

*Retrieves detailed information about a specific market*

**Arguments:**

- marketId: String



Returns: [Market](#)

### **getHighlightedVisualImageEvents**

*Retrieves events with visual images that are highlighted*

Arguments:

Returns: [Event](#), [Event]

## **betting**

### **getFreebet**

*Retrieves information about available freebets for the user*

Arguments:

Returns: [FreebetResponse](#)

### **getAllowedBetTypes**

*Retrieves allowed bet types for given selections*

Arguments:

- betTicketSelections: [BetTicketSelection]

Returns: [BetTicketSelection](#), [BetType](#), [BetType]

### **calculatePotentialReturn**

*Calculates potential return for a bet ticket before placing the bet*

Arguments:

- betTicket: [BetTicket](#)

Returns: BetslipPotentialReturn

### **getWonBetsHistory**

*Retrieves history of won bets with optional date filtering*

Arguments:

- startDate: String?
- endDate: String?
- pageIndex: Int

Returns: [BettingHistory](#)

### **cashoutBet**

*Performs a cashout operation on a specific bet*

**Arguments:**

- cashoutValue: Double
- betId: String
- stakeValue: Double?

**Returns:** [CashoutResult](#)

**getTicketSelection**

*Retrieves a specific ticket selection by its ID*

**Arguments:**

- ticketSelectionId: String

**Returns:** [TicketSelection](#)

**getSharedTicket**

*Retrieves a shared bet ticket by its ID*

**Arguments:**

- betslipId: String

**Returns:** [SharedTicketResponse](#)

**getBetHistory**

*Retrieves betting history with pagination*

**Arguments:**

- pageIndex: Int

**Returns:** [BettingHistory](#)

**calculateBetBuilderPotentialReturn**

*Calculates potential return for a bet builder ticket*

**Arguments:**

- betTicket: [BetTicket](#)

**Returns:** [BetBuilderPotentialReturn](#)

**updateBetslipSettings**

*Updates the user's betslip settings*

**Arguments:**

- betslipSettings: [BetslipSettings](#)

**Returns:** [BetslipSettings](#)

## **confirmBoostedBet**

*Confirms a boosted bet offer*

### **Arguments:**

- identifier: String

**Returns:** Bool

## **rejectBoostedBet**

*Rejects a boosted bet offer*

### **Arguments:**

- identifier: String

**Returns:** Bool

## **calculateCashout**

*Calculates the cashout value for a specific bet*

### **Arguments:**

- betId: String
- stakeValue: String?

**Returns:** [Cashout](#)

## **calculateCashback**

*Calculates potential cashback for a bet ticket*

### **Arguments:**

- betTicket: [BetTicket](#)

**Returns:** [CashbackResult](#)

## **getResolvedBetsHistory**

*Retrieves history of resolved bets with optional date filtering*

### **Arguments:**

- pageIndex: Int
- startDate: String?
- endDate: String?

**Returns:** [BettingHistory](#)

## **getOpenBetsHistory**

*Retrieves history of open bets with optional date filtering*

**Arguments:**

- startDate: String?
- pageIndex: Int
- endDate: String?

**Returns:** [BettingHistory](#)

**allowedCashoutBetIds**

*Retrieves IDs of bets that are eligible for cashout*

**Arguments:**

**Returns:** String, [String]

**placeBetBuilderBet**

*Places a bet builder bet with calculated odds*

**Arguments:**

- betTicket: [BetTicket](#)
- calculatedOdd: Double

**Returns:** [PlacedBetsResponse](#)

**getBetDetails**

*Retrieves detailed information about a specific bet*

**Arguments:**

- identifier: String

**Returns:** [Bet](#)

**placeBets**

*Places one or more bets using the provided bet tickets*

**Arguments:**

- betTickets: [BetTicket]
- useFreebetBalance: Bool

**Returns:** [BetTicket](#), [PlacedBetsResponse](#)

**getBetslipSettings**

*Retrieves the current betslip settings for the user*

**Arguments:**

**Returns:** [BetslipSettings](#)

## identity\_verification

### getSumsubAccessToken

*Retrieves an access token for Sumsub identity verification service*

**Arguments:**

- levelName: String
- userId: String

**Returns:** [AccessTokenResponse](#)

### checkDocumentationData

*Checks the status of user's submitted documentation*

**Arguments:**

**Returns:** [ApplicantDataResponse](#)

### generateDocumentTypeToken

*Generates a token for uploading a specific type of document*

**Arguments:**

- docType: String

**Returns:** [AccessTokenResponse](#)

### getSumsubApplicantData

*Retrieves applicant verification data from Sumsub*

**Arguments:**

- userId: String

**Returns:** [ApplicantDataResponse](#)

## account\_management

### updatePassword

*Updates the user's password*

**Arguments:**

- newPassword: String
- oldPassword: String

**Returns:** Bool

### updateExtraInfo

*Updates additional user information*

**Arguments:**

- address2: String?
- placeOfBirth: String?

**Returns:** [BasicResponse](#)

**updateDeviceIdentifier**

*Updates the device identifier and app version for the user*

**Arguments:**

- deviceIdentifier: String
- appVersion: String

**Returns:** [BasicResponse](#)

**verifyMobileCode**

*Verifies a mobile verification code*

**Arguments:**

- code: String
- requestId: String

**Returns:** [MobileVerifyResponse](#)

**getMobileVerificationCode**

*Requests a verification code for a mobile number*

**Arguments:**

- mobileNumber: String

**Returns:** [MobileVerifyResponse](#)

**forgotPassword**

*Initiates the password recovery process*

**Arguments:**

- secretAnswer: String?
- email: String
- secretQuestion: String?

**Returns:** Bool

**lockPlayer**

*Locks a player's account with specified duration*

**Arguments:**

- lockPeriodUnit: String?
- isPermanent: Bool?
- lockPeriod: String?

**Returns:** [BasicResponse](#)

**updateUserProfile**

*Updates the user's profile information*

**Arguments:**

- form: UpdateUserProfileForm

**Returns:** Bool

**payments****getPayments**

*Retrieves available payment methods for deposits*

**Arguments:**

**Returns:** SimplePaymentMethodsResponse

**cancelWithdrawal**

*Cancels a pending withdrawal transaction*

**Arguments:**

- paymentId: Int

**Returns:** [CancelWithdrawalResponse](#)

**cancelDeposit**

*Cancels a pending deposit transaction*

**Arguments:**

- paymentId: String

**Returns:** [BasicResponse](#)

**getPendingWithdrawals**

*Retrieves list of pending withdrawal transactions*

**Arguments:**

**Returns:** [PendingWithdrawal](#), [PendingWithdrawal]

## **prepareWithdrawal**

*Prepares a withdrawal request for processing*

### **Arguments:**

- paymentMethod: String

**Returns:** [PrepareWithdrawalResponse](#)

## **checkPaymentStatus**

*Checks the status of a payment transaction*

### **Arguments:**

- paymentId: String
- paymentMethod: String

**Returns:** [PaymentStatusResponse](#)

## **getWithdrawalMethods**

*Retrieves available withdrawal methods*

### **Arguments:**

**Returns:** [WithdrawalMethod](#), [WithdrawalMethod]

## **addPaymentInformation**

*Adds new payment information for the user*

### **Arguments:**

- fields: String
- type: String

**Returns:** [AddPaymentInformationResponse](#)

## **getTransactionsHistory**

*Retrieves transaction history for a specified date range*

### **Arguments:**

- pageNumber: Int?
- startDate: String
- endDate: String
- transactionTypes: [TransactionType]?

**Returns:** [TransactionDetail](#), TransactionType, [TransactionDetail]

## **processDeposit**

*Processes a deposit request*



**Arguments:**

- amount: Double
- paymentMethod: String
- option: String

**Returns:** [ProcessDepositResponse](#)

**processWithdrawal**

*Processes a withdrawal request*

**Arguments:**

- amount: Double
- conversionId: String?
- paymentMethod: String

**Returns:** [ProcessWithdrawalResponse](#)

**updatePayment**

*Updates payment information for an existing payment*

**Arguments:**

- encryptedCardNumber: String?
- nameOnCard: String?
- paymentId: String
- encryptedExpiryYear: String?
- returnUrl: String?
- type: String
- encryptedExpiryMonth: String?
- amount: Double
- encryptedSecurityCode: String?

**Returns:** [UpdatePaymentResponse](#)

**getPaymentInformation**

*Retrieves saved payment information for the user*

**Arguments:**

**Returns:** [PaymentInformation](#)

**wallet****getUserBalance**

*Retrieves the user's wallet balance information*

**Arguments:**

**Returns:** UserWallet

## **getUserCashbackBalance**

*Retrieves the user's cashback balance information*

**Arguments:**

**Returns:** [CashbackBalance](#)

## **authentication**

### **logout**

*Logs out the current user and invalidates their session*

**Arguments:**

**Returns:** [BasicResponse](#)

### **login**

*Authenticates a user with username and password*

**Arguments:**

- username: String
- password: String

**Returns:** UserProfile

### **getPasswordPolicy**

*Retrieves the password policy requirements*

**Arguments:**

**Returns:** PasswordPolicy

## **referral**

### **getReferees**

*Retrieves the list of users referred by the current user*

**Arguments:**

**Returns:** [Referee](#), [Referee]

### **getReferralLink**

*Retrieves the user's referral link*

**Arguments:**

**Returns:** [ReferralLink](#)

## favorites

### getPromotedBetslips

*Retrieves promoted betsliips, optionally filtered by user*

#### Arguments:

- userId: String?

**Returns:** [PromotedBetslip](#), [PromotedBetslip]

### getFavoritesList

*Retrieves all favorite lists for the current user*

#### Arguments:

**Returns:** [FavoritesListResponse](#)

### addFavoritesList

*Creates a new favorites list with the specified name*

#### Arguments:

- name: String

**Returns:** [FavoritesListAddResponse](#)

### deleteFavoriteFromList

*Deletes a favorite event from a list*

#### Arguments:

- eventId: Int

**Returns:** [FavoritesListDeleteResponse](#)

### getFavoritesFromList

*Retrieves all favorite events from a specified list*

#### Arguments:

- listId: Int

**Returns:** [FavoriteEventResponse](#)

### deleteFavoritesList

*Deletes a favorites list with the specified ID*

#### Arguments:

- listId: Int

**Returns:** [FavoritesListDeleteResponse](#)

### **addFavoriteToList**

*Adds an event to a specified favorites list*

**Arguments:**

- eventId: String
- listId: Int

**Returns:** [FavoriteAddResponse](#)

## **responsible\_gaming**

### **updateResponsibleGamingLimits**

*Updates the user's responsible gaming limits*

**Arguments:**

- newLimit: Double
- limitType: String
- hasRollingWeeklyLimits: Bool

**Returns:** Bool

### **getPersonalDepositLimits**

*Retrieves the user's personal deposit limits*

**Arguments:**

**Returns:** [PersonalDepositLimitResponse](#)

### **getResponsibleGamingLimits**

*Retrieves responsible gaming limits for specified period and limit types*

**Arguments:**

- periodTypes: String?
- limitTypes: String?

**Returns:** [ResponsibleGamingLimitsResponse](#)

### **updateWeeklyDepositLimits**

*Updates the user's weekly deposit limits*

**Arguments:**

- newLimit: Double

**Returns:** Bool

## **updateWeeklyBettingLimits**

*Updates the user's weekly betting limits*

### **Arguments:**

- newLimit: Double

**Returns:** Bool

## **getLimits**

*Retrieves all user limits information*

### **Arguments:**

**Returns:** [LimitsResponse](#)

## **documents**

### **getDocumentTypes**

*Retrieves available document types for verification*

### **Arguments:**

**Returns:** [DocumentTypesResponse](#)

### **getUserDocuments**

*Retrieves user's uploaded documents*

### **Arguments:**

**Returns:** [UserDocumentsResponse](#)

### **uploadUserDocument**

*Uploads a single user verification document*

### **Arguments:**

- documentType: String
- file: Data
- fileName: String

**Returns:** [UploadDocumentResponse](#)

### **uploadMultipleUserDocuments**

*Uploads multiple user verification documents*

### **Arguments:**

- files: [String: Data]
- documentType: String

Returns: [UploadDocumentResponse](#)

## consent\_management

### getAllConsents

*Retrieves all available consent types and their information*

**Arguments:**

**Returns:** ConsentInfo, [ConsentInfo]

### setUserConsents

*Updates user consent statuses for specified consent versions*

**Arguments:**

- unconsenVersionIds: [Int]?
- consentVersionIds: [Int]?

**Returns:** [BasicResponse](#), Int

### getUserConsents

*Retrieves the user's current consent statuses*

**Arguments:**

**Returns:** [UserConsent](#), [UserConsent]

## Real-time Services

*These services provide real-time updates through WebSocket connections.*

### subscribePreLiveMatches

*Subscribes to pre-live (upcoming) matches for a specific sport type with optional date range and sorting parameters*

**Update Information:**

- Frequency: on-change

### subscribeLiveMatches

*Subscribes to live matches updates for a specific sport type through WebSocket connection*

**Update Information:**

- Frequency: real-time

### subscribeToMarketDetails

*Subscribes to real-time updates for a specific market within an event*

**Update Information:**

- Frequency: real-time

**subscribePreLiveSportTypes**

*Subscribes to updates for available pre-live sport types within a specified date range*

**Update Information:**

- Frequency: on-change

**subscribeAllSportTypes**

*Subscribes to updates for all available sport types, including both live and pre-live sports*

**Update Information:**

- Frequency: on-change

**subscribeOutrightMarkets**

*Subscribes to outright market updates for a specific market group (e.g., tournament winner, top scorer)*

**Update Information:**

- Frequency: on-change

**subscribeEventMarkets**

*Subscribes to all markets associated with a specific event, including odds updates and market status changes*

**Update Information:**

- Frequency: real-time

**subscribeLiveSportTypes**

*Subscribes to updates for currently live sport types and their active events*

**Update Information:**

- Frequency: real-time

**subscribeToLiveDataUpdates**

*Subscribes to real-time live data updates for a specific event, including detailed statistics and play-by-play information*

**Update Information:**

- Frequency: real-time

**subscribeEventDetails**

*Subscribes to detailed updates for a specific event, including scores, statistics, and market information*

**Update Information:**

- Frequency: real-time
- Includes:
  - score updates
  - match statistics
  - timeline events
  - market information
  - event status changes

**subscribeCompetitionMatches**

*Subscribes to matches updates for a specific competition identified by its market group ID*

**Update Information:**

- Frequency: real-time

## Data Models

*This section describes the data structures used in the API.*

### Ⓜ AccessTokenResponse

**Properties:**

Name	Type
token	String?
userId	String?
description	String?
code	Int?

### Ⓜ ActivePlayerServe

**Properties:**

Name	Type
home	ActivePlayerServe
away	ActivePlayerServe

**Related Models:**

- [ActivePlayerServe](#)

### Ⓜ AddPaymentInformationResponse

**Properties:**

Name	Type
status	String



message String?

## Ⓜ ApplicantDataInfo

### Properties:

Name	Type
applicantDocs	[ApplicantDoc]?

### Related Models:

- [ApplicantDoc](#)

## Ⓜ ApplicantDataResponse

### Properties:

Name	Type
externalUserId	String?
info	ApplicantDataInfo?
reviewData	ApplicantReviewData?
description	String?

### Related Models:

- [ApplicantDataInfo](#)
- [ApplicantReviewData](#)

## Ⓜ ApplicantDoc

### Properties:

Name	Type
docType	String

## Ⓜ ApplicantReviewData

### Properties:

Name	Type
attemptCount	Int
createDate	String
reviewDate	String?
reviewResult	ApplicantReviewResult?
reviewStatus	String
levelName	String

### Related Models:

- [ApplicantReviewResult](#)

## Ⓜ ApplicantReviewResult

**Properties:**

Name	Type
reviewAnswer	String
reviewRejectType	String?
moderationComment	String?

**Ⓜ ApplicantRootResponse****Properties:**

Name	Type
status	String
message	String?
data	ApplicantDataResponse

**Related Models:**

- [ApplicantDataResponse](#)

**Ⓜ AvailableBonus****Properties:**

Name	Type
id	String
bonusPlanId	Int
name	String
description	String?
type	String
amount	Double
triggerDate	String
expiryDate	String
wagerRequirement	Double?
imageUrl	String?

**Ⓜ AvailableBonusResponse****Properties:**

Name	Type
status	String
bonuses	[AvailableBonus]

**Related Models:**

- [AvailableBonus](#)

**Ⓜ BalanceResponse****Properties:**

Name	Type
status	String
message	String?
currency	String?
loyaltyPoint	Int?
vipStatus	String?
totalBalance	String?
totalBalanceNumber	Double?
withdrawableBalance	String?
withdrawableBalanceNumber	Double?
bonusBalance	String?
bonusBalanceNumber	Double?
pendingBonusBalance	String?
pendingBonusBalanceNumber	Double?
casinoPlayableBonusBalance	String?
casinoPlayableBonusBalanceNumber	Double?
sportsbookPlayableBonusBalance	String?
sportsbookPlayableBonusBalanceNumber	Double?
withdrawableEscrowBalance	String?
withdrawableEscrowBalanceNumber	Double?
totalWithdrawableBalance	String?
totalWithdrawableBalanceNumber	Double?
withdrawRestrictionAmount	String?
withdrawRestrictionAmountNumber	Double?
totalEscrowBalance	String?
totalEscrowBalanceNumber	Double?

## Ⓜ BankPaymentDetail

### Properties:

Name	Type
id	Int
paymentInfold	Int
key	String
value	String

## Ⓜ BankPaymentInfo

### Properties:

Name	Type
id	Int
partyId	Int
type	String
description	String?
priority	Int?

details [BankPaymentDetail]

#### Related Models:

- [BankPaymentDetail](#)

### Ⓜ Banner

#### Properties:

Name	Type
id	String
name	String
title	String
imageUrl	String
bodyText	String?
type	String
linkUrl	String?
marketId	String?

### Ⓜ BannerResponse

#### Properties:

Name	Type
bannerItems	[Banner]

#### Related Models:

- [Banner](#)

### Ⓜ BannerSpecialAction

#### Properties:

Name	Type
register	BannerSpecialAction
none	BannerSpecialAction

#### Related Models:

- [BannerSpecialAction](#)

### Ⓜ BasicResponse

#### Properties:

Name	Type
status	String
message	String?

### Ⓜ Bet

**Properties:**

Name	Type
identifier	String
eventName	String
homeTeamName	String?
awayTeamName	String?
sportTypeName	String
type	String
state	BetState
result	BetResult
globalState	BetState
marketName	String
outcomeName	String
eventResult	String?
potentialReturn	Double?
totalReturn	Double?
totalOdd	Double
totalStake	Double
attemptedDate	Date
oddNumerator	Double
oddDenominator	Double
order	Int
eventId	Double
eventDate	Date?
tournamentCountryName	String?
tournamentName	String?
freeBet	Bool
partialCashoutReturn	Double?
partialCashoutStake	Double?
betslipId	Int?
cashbackReturn	Double?
freebetReturn	Double?
potentialCashbackReturn	Double?
potentialFreebetReturn	Double?
dateFormatter	DateFormatter
fallbackDateFormatter	DateFormatter

**Related Models:**

- [BetState](#)
- [BetResult](#)

**Ⓜ BetBuilderPotentialReturn****Properties:**

Name	Type
------	------

potentialReturn Double  
calculatedOdds Double

## Ⓜ BetResult

### Properties:

Name	Type
won	BetResult
lost	BetResult
drawn	BetResult
open	BetResult
void	BetResult
pending	BetResult
notSpecified	BetResult

### Related Models:

- [BetResult](#)

## Ⓜ BetSlip

### Properties:

Name	Type
tickets	[BetTicket]

### Related Models:

- [BetTicket](#)

## Ⓜ BetSlipStateResponse

### Properties:

Name	Type
tickets	[BetTicket]

### Related Models:

- [BetTicket](#)

## Ⓜ BetState

### Properties:

Name	Type
attempted	BetState
opened	BetState
closed	BetState
settled	BetState
cancelled	BetState

allStates	BetState
won	BetState
lost	BetState
cashedException	BetState
void	BetState
undefined	BetState

#### Related Models:

- [BetState](#)

### Ⓜ BetTicket

#### Properties:

Name	Type
selections	[BetTicketSelection]
betTypeCode	String
winStake	String
potentialReturn	Double?
pool	Bool

#### Related Models:

- [BetTicketSelection](#)

### Ⓜ BetTicketSelection

#### Properties:

Name	Type
identifier	String
eachWayReduction	String
eachWayPlaceTerms	String
idFOPriceType	String
isTrap	String
priceUp	String
priceDown	String

### Ⓜ BetType

#### Properties:

Name	Type
typeCode	String
typeName	String
potencialReturn	Double
totalStake	Double
numberOfIndividualBets	Int

## Ⓜ BetslipPotentialReturnResponse

### Properties:

Name	Type
potentialReturn	Double
totalStake	Double
numberOfBets	Int
totalOdd	Double?

## Ⓜ BetslipSettings

### Properties:

Name	Type
oddChangeLegacy	BetslipOddChangeSetting?
oddChangeRunningOrPreMatch	BetslipOddChangeSetting?

### Related Models:

## Ⓜ BettingHistory

### Properties:

Name	Type
bets	[Bet]

### Related Models:

- [Bet](#)

## Ⓜ CancelWithdrawalResponse

### Properties:

Name	Type
status	String
amount	String
currency	String

## Ⓜ CashbackBalance

### Properties:

Name	Type
status	String
balance	String?
message	String?

## Ⓜ CashbackResult

### Properties:



Name	Type
id	Double
amount	Double?
amountFree	Double?

## Ⓜ Cashout

### Properties:

Name	Type
cashoutValue	Double
partialCashoutAvailable	Bool?

## Ⓜ CashoutResult

### Properties:

Name	Type
cashoutResult	Int?
cashoutReoffer	Double?
message	String?

## Ⓜ CheckCredentialResponse

### Properties:

Name	Type
status	String
exists	String
fieldExist	Bool

## Ⓜ CheckUsernameResponse

### Properties:

Name	Type
errors	[CheckUsernameError]?
status	String
message	String?
additionalInfos	[CheckUsernameAdditionalInfo]?

### Related Models:

## Ⓜ CompetitionMarketGroup

### Properties:

Name	Type
id	String
name	String
events	[Event]

#### Related Models:

- [Event](#)

### Ⓜ CompetitionParentNode

#### Properties:

Name	Type
id	String
name	String
categoryName	String

### Ⓜ ConfirmBetPlaceResponse

#### Properties:

Name	Type
state	Int
detailedState	Int
statusCode	String?
statusText	String?

### Ⓜ Consent

#### Properties:

Name	Type
id	Int
key	String
name	String
consentVersionId	Int
status	String?
isMandatory	Bool?

### Ⓜ ConsentsResponse

#### Properties:

Name	Type
status	String
consents	[Consent]

#### Related Models:

- [Consent](#)

### Ⓜ ContentContainer

#### Properties:

Name	Type
------	------

liveEvents	ContentContainer
preLiveEvents	ContentContainer
liveSports	ContentContainer
preLiveSports	ContentContainer
allSports	ContentContainer
eventDetails	ContentContainer
eventDetailsLiveData	ContentContainer
eventGroup	ContentContainer
outrightEventGroup	ContentContainer
eventSummary	ContentContainer
marketDetails	ContentContainer
updateEventSecondaryMarkets	ContentContainer
updateEventMainMarket	ContentContainer
addEvent	ContentContainer
addMarket	ContentContainer
addSelection	ContentContainer
addSport	ContentContainer
removeEvent	ContentContainer
removeMarket	ContentContainer
removeSelection	ContentContainer
removeSport	ContentContainer
enableMarket	ContentContainer
updateEventLiveDataExtended	ContentContainer
updateAllSportsLiveCount	ContentContainer
updateAllSportsEventCount	ContentContainer
updateEventState	ContentContainer
updateEventTime	ContentContainer
updateEventScore	ContentContainer
updateActivePlayer	ContentContainer
updateEventDetailedScore	ContentContainer
updateMarketTradability	ContentContainer
updateEventMarketCount	ContentContainer
updateOutcomeOdd	ContentContainer
updateOutcomeTradability	ContentContainer
unknown	ContentContainer
contentIdentifier	ContentIdentifier?

#### Related Models:

- [ContentContainer](#)

### Ⓜ CountryInfo

#### Properties:

Name	Type
name	String

iso2Code String  
phonePrefix String

## Ⓜ DepositMethod

### Properties:

Name	Type
code	String
paymentMethod	String
methods	[PaymentMethod]?

### Related Models:

- [PaymentMethod](#)

## Ⓜ DocumentType

### Properties:

Name	Type
documentType	String
issueDateRequired	Bool?
expiryDateRequired	Bool?
documentNumberRequired	Bool?
multipleFileRequired	Bool?

## Ⓜ DocumentTypesResponse

### Properties:

Name	Type
status	String
documentTypes	[DocumentType]

### Related Models:

- [DocumentType](#)

## Ⓜ Event

### Properties:

Name	Type
id	String
homeName	String?
awayName	String?
sportTypeName	String?
sportTypeCode	String?
sportIdCode	String?
competitionId	String?

competitionName	String?
startDate	Date?
markets	[Market]
tournamentCountryName	String?
numberMarkets	Int?
name	String?
homeScore	Int
awayScore	Int
matchTime	String?
status	EventStatus
scores	[String: Score]
activePlayerServing	ActivePlayerServe?
trackableReference	String?
dateFormatter	DateFormatter
fallbackDateFormatter	DateFormatter

#### Related Models:

- [Market](#)
- [EventStatus](#)
- [Score](#)
- [ActivePlayerServe](#)

### Ⓜ EventLiveDataExtended

#### Properties:

Name	Type
id	String
homeScore	Int?
awayScore	Int?
matchTime	String?
status	EventStatus?
scores	[String: Score]
activePlayerServing	ActivePlayerServe?

#### Related Models:

- [EventStatus](#)
- [Score](#)
- [ActivePlayerServe](#)

### Ⓜ EventStatus

#### Properties:

Name	Type
unknown	EventStatus
notStarted	EventStatus
inProgress	EventStatus

ended      EventStatus  
stringValue String

**Related Models:**

- [EventStatus](#)

**Ⓜ EventsGroup**

**Properties:**

Name	Type
events	[Event]
marketGroupld	String?

**Related Models:**

- [Event](#)

**Ⓜ FavoriteAddResponse**

**Properties:**

Name	Type
displayOrder	Int?
idAccountFavorite	Int?

**Ⓜ FavoriteEvent**

**Properties:**

Name	Type
id	String
name	String
favoriteListld	Int
accountFavoriteld	Int

**Ⓜ FavoriteEventResponse**

**Properties:**

Name	Type
favoriteEvents	[FavoriteEvent]

**Related Models:**

- [FavoriteEvent](#)

**Ⓜ FavoriteList**

**Properties:**

Name	Type
------	------

id Int  
name String  
customerId Int

## Ⓜ FavoritesListAddResponse

### Properties:

#### Name Type

listId Int

## Ⓜ FavoritesListDeleteResponse

### Properties:

#### Name Type

listId String?

## Ⓜ FavoritesListResponse

### Properties:

Name	Type
favoritesList	[FavoriteList]

### Related Models:

- [FavoriteList](#)

## Ⓜ FieldError

### Properties:

#### Name Type

field String  
error String

## Ⓜ FreebetResponse

### Properties:

Name	Type
balance	Double

## Ⓜ GetCountriesResponse

### Properties:

Name	Type
status	String
countries	[String]

## Ⓜ GetCountryInfoResponse

**Properties:**

Name	Type
status	String
countryInfo	CountryInfo

**Related Models:**

- [CountryInfo](#)

**Ⓜ GrantedBonus****Properties:**

Name	Type
id	Int
name	String
status	String
amount	String
triggerDate	String
expiryDate	String
wagerRequirement	String?
amountWagered	String?

**Ⓜ GrantedBonusResponse****Properties:**

Name	Type
status	String
bonuses	[GrantedBonus]

**Related Models:**

- [GrantedBonus](#)

**Ⓜ HeadlineItem****Properties:**

Name	Type
idfwheadline	String?
marketGroupId	String?
marketId	String?
name	String?
title	String?
tsactivefrom	String?
tsactiveto	String?
idfwheadlinetype	String?
headlinemediatype	String?
categoryName	String?



numofselections	String?
imageUrl	String?
linkURL	String?
oldMarketId	String?
tournamentCountryName	String?

## Ⓜ **HeadlineResponse**

### **Properties:**

Name	Type
headlineItems	[HeadlineItem]?

### **Related Models:**

- [HeadlineItem](#)

## Ⓜ **HighlightedEventPointer**

### **Properties:**

Name	Type
status	String
sportId	String
eventId	String
eventType	String?
countryId	String

## Ⓜ **KYCStatusDetail**

### **Properties:**

Name	Type
expiryDate	String?

## Ⓜ **LimitPending**

### **Properties:**

Name	Type
effectiveDate	String
limit	String
limitNumber	Double

## Ⓜ **LimitsResponse**

### **Properties:**

Name	Type
status	String
wagerLimit	String?

lossLimit String?  
currency String  
pendingWagerLimit LimitPending?

**Related Models:**

- [LimitPending](#)

**Ⓜ LoginResponse**

**Properties:**

Name	Type
status	String
partyId	String?
username	String?
language	String?
currency	String?
email	String?
sessionKey	String?
parentId	String?
level	String?
userType	String?
isFirstLogin	String?
registrationStatus	String?
pendingLimitConfirmation	String?
country	String?
kycStatus	String?
lockStatus	String?
securityVerificationRequiredFields	[String]?
message	String?
lockUntilDateFormatted	String?
kycStatusDetails	KYCStatusDetail?

**Related Models:**

- [KYCStatusDetail](#)

**Ⓜ Market**

**Properties:**

Name	Type
id	String
name	String
outcomes	[Outcome]
marketTypeId	String?
eventMarketTypeId	String?
marketTypeCategoryId	String?

eventName	String?
isMainOutright	Bool?
eventMarketCount	Int?
isTradable	Bool
startDate	Date?
homeParticipant	String?
awayParticipant	String?
eventId	String?
isOverUnder	Bool
marketDigitLine	String?
outcomesOrder	OutcomesOrder
competitionId	String?
competitionName	String?
sportTypeName	String?
sportTypeCode	String?
sportIdCode	String?
tournamentCountryName	String?
customBetAvailable	Bool?
isMainMarket	Bool

#### Related Models:

- [Outcome](#)

### Ⓜ MarketGroup

#### Properties:

Name	Type
markets	[Market]
marketGroupId	String?

#### Related Models:

- [Market](#)

### Ⓜ MarketGroupPromotedSport

#### Properties:

Name	Type
id	String
typeId	String?
name	String?

### Ⓜ MobileVerifyResponse

#### Properties:

Name	Type
------	------

status String  
message String?  
requestId Int?

## Ⓜ NotificationType

### Properties:

Name	Type
listeningStarted	NotificationType
contentChanges	NotificationType
subscriberIdNotFoundError	NotificationType
genericError	NotificationType
unknown	NotificationType

### Related Models:

- [NotificationType](#)

## Ⓜ OpenSessionResponse

### Properties:

Name	Type
status	String
launchToken	String

## Ⓜ Outcome

### Properties:

Name	Type
id	String
name	String
hashCode	String
marketId	String?
orderValue	String?
externalReference	String?
odd	OddFormat
priceNumerator	String?
priceDenominator	String?
isTradable	Bool?
isTerminated	Bool?
isOverUnder	Bool
customBetAvailableMarket	Bool?

### Related Models:

## Ⓜ PaymentInformation

### Properties:

Name	Type
status	String
data	[BankPaymentInfo]

### Related Models:

- [BankPaymentInfo](#)

## Ⓜ PaymentMethod

### Properties:

Name	Type
name	String
type	String
brands	[String]?

## Ⓜ PaymentStatusResponse

### Properties:

Name	Type
status	String
paymentId	String?
paymentStatus	String?
message	String?

## Ⓜ PaymentsResponse

### Properties:

Name	Type
status	String
depositMethods	[DepositMethod]

### Related Models:

- [DepositMethod](#)

## Ⓜ PendingWithdrawal

### Properties:

Name	Type
status	String
paymentId	Int
amount	String

## Ⓜ PendingWithdrawalResponse

**Properties:**

Name	Type
status	String
pendingWithdrawals	[PendingWithdrawal]

**Related Models:**

- [PendingWithdrawal](#)

**Ⓜ PersonalDepositLimitResponse****Properties:**

Name	Type
status	String
dailyLimit	String?
weeklyLimit	String?
monthlyLimit	String?
currency	String
hasPendingWeeklyLimit	String?
pendingWeeklyLimit	String?
pendingWeeklyLimitEffectiveDate	String?

**Ⓜ PlacedBetEntry****Properties:**

Name	Type
identifier	String
potentialReturn	Double
totalAvailableStake	Double
betLegs	[PlacedBetLeg]
type	String?

**Related Models:**

- [PlacedBetLeg](#)

**Ⓜ PlacedBetLeg****Properties:**

Name	Type
identifier	String
priceType	String
odd	Double
priceNumerator	Int
priceDenominator	Int

**Ⓜ PlacedBetsResponse**

**Properties:**

Name	Type
identifier	String
responseCode	String
detailedResponseCode	String?
errorMessage	String?
totalStake	Double
bets	[PlacedBetEntry]

**Related Models:**

- [PlacedBetEntry](#)

**Ⓜ PlayerInfoResponse****Properties:**

Name	Type
status	String
partyId	String
userId	String
email	String
firstName	String?
lastName	String?
middleName	String?
nickname	String?
language	String?
phone	String?
phoneCountryCode	String?
phoneLocalNumber	String?
phoneNeedsReview	Bool?
birthDate	String?
birthDateFormatted	Date
regDate	String?
regDateFormatted	Date?
mobilePhone	String?
mobileCountryCode	String?
mobileLocalNumber	String?
mobileNeedsReview	Bool?
currency	String?
lastLogin	String?
lastLoginFormatted	Date?
level	Int?
parentID	String?
userType	Int?
isAutopay	Bool?
registrationStatus	String?

sessionKey	String?
vipStatus	String?
kycStatus	String?
emailVerificationStatus	String
verificationStatus	String?
lockedStatus	String?
gender	String?
contactPreference	String?
verificationMethod	String?
docNumber	String?
readonlyFields	String?
accountNumber	String?
idCardNumber	String?
madeDeposit	Bool?
testPlayer	Bool?
address	String?
city	String?
province	String?
postalCode	String?
country	String?
nationality	String?
municipality	String?
streetNumber	String?
building	String?
unit	String?
floorNumber	String?
birthDepartment	String?
birthCity	String?
birthCountryCode	String?
extraInfos	[ExtraInfo]?

#### Related Models:

#### Ⓜ PrepareWithdrawalResponse

##### Properties:

Name	Type
status	String
conversionId	String?
message	String?

#### Ⓜ ProcessDepositResponse

##### Properties:

Name	Type
status	String



paymentId String?  
continueUrl String?  
clientKey String?  
sessionId String?  
sessionData String?  
message String?

## Ⓜ ProcessWithdrawalResponse

### Properties:

Name	Type
status	String
paymentId	String?
message	String?

## Ⓜ PromotedBetslip

### Properties:

Name	Type
selections	[PromotedBetslipSelection]
betslipCount	Int

### Related Models:

- [PromotedBetslipSelection](#)

## Ⓜ PromotedBetslipSelection

### Properties:

Name	Type
id	String?
beginDate	String?
betOfferId	Int?
country	String?
countryId	String?
eventId	String?
eventType	String?
league	String?
leagueId	String?
market	String?
marketId	Int?
marketType	String?
marketTypeId	Int?
orakoEventId	String
orakoMarketId	String
orakoSelectionId	String

outcomeType	String?
outcomeId	Int?
participantIds	[String]?
participants	[String]?
period	String?
periodId	Int?
quote	Double?
quoteGroup	String?
sport	String?
sportId	String?
status	String?

## Ⓜ PromotedBetslipsBatchResponse

### Properties:

Name	Type
promotedBetslips	[PromotedBetslip]
status	String

### Related Models:

- [PromotedBetslip](#)

## Ⓜ PromotedBetslipsInternalRequest

### Properties:

Name	Type
body	VaixBatchBody
name	String
statusCode	Int

### Related Models:

- [VaixBatchBody](#)

## Ⓜ PromotedSport

### Properties:

Name	Type
id	String
name	String
marketGroups	[MarketGroupPromotedSport]

### Related Models:

- [MarketGroupPromotedSport](#)

## Ⓜ PromotedSportsNodeResponse

**Properties:**

Name	Type
promotedSports	[PromotedSport]

**Related Models:**

- [PromotedSport](#)

**Ⓜ PromotedSportsResponse****Properties:**

Name	Type
promotedSports	[PromotedSport]

**Related Models:**

- [PromotedSport](#)

**Ⓜ PromotionalBanner****Properties:**

Name	Type
id	String
name	String?
bannerType	String?
imageUrl	String?
bannerDisplay	String?
linkType	String?
location	String?
bannerContents	[String]?

**Ⓜ PromotionalBannersResponse****Properties:**

Name	Type
promotionalBannerItems	[PromotionalBanner]

**Related Models:**

- [PromotionalBanner](#)

**Ⓜ PromotionalStoriesResponse****Properties:**

Name	Type
promotionalStories	[PromotionalStory]

**Related Models:**

- [PromotionalStory](#)

## Ⓜ PromotionalStory

### Properties:

Name	Type
id	String
title	String
imageUrl	String
linkUrl	String
bodyText	String

## Ⓜ RedeemBonus

### Properties:

Name	Type
id	Int
name	String
status	String
triggerDate	String
expiryDate	String
amount	String
wagerRequired	String
amountWagered	String

## Ⓜ RedeemBonusResponse

### Properties:

Name	Type
status	String
message	String?
bonus	RedeemBonus?

### Related Models:

- [RedeemBonus](#)

## Ⓜ Referee

### Properties:

Name	Type
id	Int
username	String
registeredAt	String
kycStatus	String
depositPassed	Bool

## Ⓜ RefereesResponse

### Properties:

Name	Type
status	String
referees	[Referee]

### Related Models:

- [Referee](#)

## Ⓜ ReferralLink

### Properties:

Name	Type
code	String
link	String

## Ⓜ ReferralResponse

### Properties:

Name	Type
status	String
referralLinks	[ReferralLink]

### Related Models:

- [ReferralLink](#)

## Ⓜ ResponsibleGamingLimit

### Properties:

Name	Type
id	Int
partyId	Int
limitType	String
periodType	String
effectiveDate	String
expiryDate	String
limit	Double

## Ⓜ ResponsibleGamingLimitsResponse

### Properties:

Name	Type
status	String
limits	[ResponsibleGamingLimit]

#### Related Models:

- [ResponsibleGamingLimit](#)

### Ⓜ ScheduledSport

#### Properties:

Name	Type
id	String
name	String

### Ⓜ Score

#### Properties:

Name	Type
set	Score
gamePart	Score
matchFull	Score
sortValue	Int
key	String

#### Related Models:

- [Score](#)

### Ⓜ ScoreCodingKeys

#### Properties:

Name	Type
gameScore	ScoreCodingKeys
currentScore	ScoreCodingKeys
matchScore	ScoreCodingKeys
periodScore	ScoreCodingKeys
setScore	ScoreCodingKeys
frameScore	ScoreCodingKeys
stringValue	String
intValue	Int?

#### Related Models:

- [ScoreCodingKeys](#)

### Ⓜ SharedBet

#### Properties:

Name	Type
betSelections	[SharedBetSelection]
winStake	Double

potentialReturn Double  
totalStake Double

**Related Models:**

- [SharedBetSelection](#)

**Ⓜ SharedBetSelection**

**Properties:**

Name	Type
id	Double
priceDenominator	Int
priceNumerator	Int
priceType	String

**Ⓜ SharedTicketResponse**

**Properties:**

Name	Type
bets	[SharedBet]
totalStake	Double
betId	Double

**Related Models:**

- [SharedBet](#)

**Ⓜ SportCompetition**

**Properties:**

Name	Type
id	String
name	String
numberEvents	String
numberOutrightEvents	String

**Ⓜ SportCompetitionInfo**

**Properties:**

Name	Type
id	String
name	String
marketGroups	[SportCompetitionMarketGroup]
numberOutrightEvents	String
numberOutrightMarkets	String
parentId	String?

#### Related Models:

- [SportCompetitionMarketGroup](#)

### Ⓜ SportCompetitionMarketGroup

#### Properties:

Name	Type
id	String
name	String

### Ⓜ SportNode

#### Properties:

Name	Type
id	String
name	String
numberEvents	Int
numberOutrightEvents	Int
numberOutrightMarkets	Int
numberLiveEvents	Int
alphaCode	String

### Ⓜ SportNodeInfo

#### Properties:

Name	Type
id	String
regionNodes	[SportRegion]
navigationTypes	[String]?
name	String?
defaultOrder	Int?
numMarkets	String?
numEvents	String?
numOutrightMarkets	String?
numOutrightEvents	String?

#### Related Models:

- [SportRegion](#)

### Ⓜ SportRadarError

#### Properties:

Name	Type
unkownSportId	SportRadarError
unkownContentId	SportRadarError



ignoredContentInitialData SportRadarError  
ignoredContentUpdate SportRadarError

**Related Models:**

- [SportRadarError](#)

**Ⓜ SportRegion**

**Properties:**

Name	Type
id	String
name	String?
numberEvents	String
numberOutrightEvents	String

**Ⓜ SportRegionInfo**

**Properties:**

Name	Type
id	String
name	String
competitionNodes	[SportCompetition]

**Related Models:**

- [SportCompetition](#)

**Ⓜ SportType**

**Properties:**

Name	Type
name	String
numericId	String?
alphaId	String?
numberEvents	Int
numberOutrightEvents	Int
numberOutrightMarkets	Int
numberLiveEvents	Int

**Ⓜ SportTypeDetails**

**Properties:**

Name	Type
sportType	SportType
eventsCount	Int
sportName	String

#### Related Models:

- [SportType](#)

### Ⓜ SportsList

#### Properties:

Name	Type
sportNodes	[SportNode]?

#### Related Models:

- [SportNode](#)

### Ⓜ StatusResponse

#### Properties:

Name	Type
status	String
errors	[FieldError]?
message	String?

#### Related Models:

- [FieldError](#)

### Ⓜ SupportRequest

#### Properties:

Name	Type
id	Int
status	String

### Ⓜ SupportResponse

#### Properties:

Name	Type
request	SupportRequest?
error	String?
description	String?

#### Related Models:

- [SupportRequest](#)

### Ⓜ TicketSelection

#### Properties:

Name	Type
------	------

id	String
marketId	String
name	String
priceDenominator	String
priceNumerator	String
odd	Double

## Ⓜ TicketSelectionResponse

### Properties:

Name	Type
data	TicketSelection?
errorType	String?

### Related Models:

- [TicketSelection](#)

## Ⓜ TopCompetitionData

### Properties:

Name	Type
title	String
competitions	[TopCompetitionPointer]

### Related Models:

- [TopCompetitionPointer](#)

## Ⓜ TopCompetitionPointer

### Properties:

Name	Type
id	String
name	String
competitionId	String

## Ⓜ TopCompetitionsResponse

### Properties:

Name	Type
data	[TopCompetitionData]

### Related Models:

- [TopCompetitionData](#)

## Ⓜ TransactionDetail

**Properties:**

Name	Type
id	Int
dateTime	String
type	String
amount	Double
postBalance	Double
amountBonus	Double
postBalanceBonus	Double
currency	String
paymentId	Int?
gameTranId	String?
reference	String?
escrowTranType	String?
escrowTranSubType	String?
escrowType	String?

**Ⓜ TransactionsHistoryResponse****Properties:**

Name	Type
status	String
transactions	[TransactionDetail]?

**Related Models:**

- [TransactionDetail](#)

**Ⓜ UpdatePaymentAction****Properties:**

Name	Type
paymentMethodType	String
url	String
method	String
type	String

**Ⓜ UpdatePaymentResponse****Properties:**

Name	Type
resultCode	String
action	UpdatePaymentAction?

**Related Models:**

- [UpdatePaymentAction](#)

## Ⓜ UploadDocumentResponse

### Properties:

Name	Type
status	String
message	String?

## Ⓜ UserConsent

### Properties:

Name	Type
consentInfo	UserConsentInfo
consentStatus	String

### Related Models:

- [UserConsentInfo](#)

## Ⓜ UserConsentInfo

### Properties:

Name	Type
id	Int
key	String
name	String
consentVersionId	Int
isMandatory	Bool?

## Ⓜ UserConsentsResponse

### Properties:

Name	Type
status	String
message	String?
userConsents	[UserConsent]

### Related Models:

- [UserConsent](#)

## Ⓜ UserDocument

### Properties:

Name	Type
documentType	String
fileName	String?
status	String

uploadDate String  
userDocumentFiles [UserDocumentFile]?

**Related Models:**

- [UserDocumentFile](#)

**Ⓜ UserDocumentFile**

**Properties:**

Name	Type
fileName	String

**Ⓜ UserDocumentsResponse**

**Properties:**

Name	Type
status	String
userDocuments	[UserDocument]

**Related Models:**

- [UserDocument](#)

**Ⓜ VaixBatchBody**

**Properties:**

Name	Type
data	VaixBatchData
status	String

**Related Models:**

- [VaixBatchData](#)

**Ⓜ VaixBatchData**

**Properties:**

Name	Type
promotedBetslips	[PromotedBetslip]
count	Int

**Related Models:**

- [PromotedBetslip](#)

**Ⓜ WithdrawalMethod**

**Properties:**

Name	Type
code	String
paymentMethod	String
minimumWithdrawal	String
maximumWithdrawal	String
conversionRequired	Bool

## Ⓜ WithdrawalMethodsResponse

### Properties:

Name	Type
status	String
withdrawalMethods	[WithdrawalMethod]

### Related Models:

- [WithdrawalMethod](#)