**CMTech IMS**

**API Specification**

**Version 2.8**

2015-08-27

CMTech

Index

[1 Version Control 9](#_Toc428525286)

[2 High Level Architecture 10](#_Toc428525287)

[3 TODO List 12](#_Toc428525288)

[3.4.1. GetAnnouncement 12](#_Toc428525289)

[4 Front API 14](#_Toc428525290)

[4.1 Player 14](#_Toc428525291)

[4.1.1 Login ( POST ) 14](#_Toc428525292)

[4.1.2 Logout ( PUT ) 16](#_Toc428525293)

[4.1.3 CheckDuplicateData ( POST ) 17](#_Toc428525294)

[4.1.4 Signup ( POST ) 17](#_Toc428525295)

[4.1.5 ExtendAccessToken ( PUT ) - unused 19](#_Toc428525299)

[4.1.6 GetPlayerDetail ( GET ) 19](#_Toc428525308)

[4.1.7 ChangePassword ( PUT ) 21](#_Toc428525309)

[4.1.8 EditPlayerDetail ( PUT ) 21](#_Toc428525310)

[4.1.9 ForgotPassword ( PUT ) 23](#_Toc428525311)

[4.1.10 ForgotNickname ( POST ) 23](#_Toc428525312)

[4.1.11 LoginToGsp ( POST ) 24](#_Toc428525313)

[4.1.12 SSOLogin ( GET ) 25](#_Toc428525314)

[4.1.13 GetGameHistory ( POST ) 27](#_Toc428525315)

[4.2 Finance 28](#_Toc428525316)

[4.2.1 RequestDeposit (POST) 28](#_Toc428525317)

[4.2.2 PSPDepositResult (PUT) 29](#_Toc428525323)

[4.2.3 RequestWithdrawal (POST) 30](#_Toc428525329)

[4.2.4 WalletTransfer (POST) 32](#_Toc428525335)

[4.2.5 GetBalance ( GET ) 32](#_Toc428525341)

[4.2.6 GetMainBalance ( GET ) 34](#_Toc428525347)

[4.2.7 GetGspBalance( GET ) 34](#_Toc428525353)

[4.2.8 GetPlayerTransactionHistory ( POST ) 35](#_Toc428525359)

[4.2.9 GetPaymentTransactionHistory ( POST ) 38](#_Toc428525369)

[4.2.10 GetGameTransactionHistory ( POST ) 40](#_Toc428525379)

[4.2.11 PSP Deposit / Withdraw => Robert 42](#_Toc428525389)

[4.3 SystemSetting 42](#_Toc428525390)

[4.3.1 GetSecurityQuestionList ( GET ) 42](#_Toc428525391)

[4.3.2 GetLanguageList ( GET ) 43](#_Toc428525392)

[4.3.3 GetAgentCountryList ( GET ) 44](#_Toc428525404)

[4.3.4 GetAgentCurrencyList (GET) 45](#_Toc428525405)

[4.3.5 GetPlayerCurrencyCodeList ( GET ) – (Unused) 45](#_Toc428525406)

[4.3.6 GetPlayerBankList ( GET ) 46](#_Toc428525407)

[4.3.7 GetAgentPspList ( GET ) 47](#_Toc428525408)

[4.3.8 GetAgentGspList ( GET ) 47](#_Toc428525409)

[4.3.9 GetAgentProductGspGameList ( GET ) 48](#_Toc428525410)

[4.3.10 CheckURL( GET ) 50](#_Toc428525411)

[4.3.11 CheckServerIP( GET ) 51](#_Toc428525412)

[4.3.12 GetCountryInfoByIP( GET ) 51](#_Toc428525413)

[4.3.13 GetGspGameList( POST ) 51](#_Toc428525414)

[4.4 Operation 55](#_Toc428525415)

[4.4.1 GetAnnouncement (POST) 55](#_Toc428525416)

[4.4.2 GetQuestionAnswerList (POST) 57](#_Toc428525417)

[4.4.3 LeaveQuestion (POST) 58](#_Toc428525418)

[4.4.4 GetAnnounceContents (GET) 59](#_Toc428525419)

[4.4.5 SendFrontWebSms (POST) 59](#_Toc428525420)

[4.5 Marketing 60](#_Toc428525421)

[4.5.1 GetPlayerCouponHistory (GET) 60](#_Toc428525422)

[4.5.2 UseCoupon (PUT) 61](#_Toc428525423)

[4.6 Agent 62](#_Toc428525424)

[4.6.1 GetAgentContactInfo (GET) 62](#_Toc428525425)

[4.7 FraudAndRisk 63](#_Toc428525426)

[4.7.1 GetAgentBlackIpList (GET) 63](#_Toc428525427)

[5 Main Wallet API 63](#_Toc428525428)

[5.1 GamePlay (GamePlay) 64](#_Toc428525429)

[5.1.1 ValidateToken ( GET ) – From W88 to IMS 66](#_Toc428525430)

[5.1.2 CreateMember (POST) 68](#_Toc428525436)

[5.1.3 GetBalance ( GET ) 69](#_Toc428525437)

[5.1.4 PlayGame 70](#_Toc428525443)

[5.1.4.1 LiveCasino 70](#_Toc428525444)

[5.1.4.2 3DSlots 71](#_Toc428525445)

[5.1.4.3 BetSoft 72](#_Toc428525446)

[5.1.4.4 Top Trend Gaming (TTG) 73](#_Toc428525447)

[5.1.4.5 CTXM 73](#_Toc428525448)

[5.1.4.6 Play N Go (PNG) 73](#_Toc428525449)

[5.1.5 PostTransfer( POST ) 73](#_Toc428525450)

[5.1.6 RollbackTransfer ( POST ) 74](#_Toc428525451)

[5.1.7 CheckTransactionResult (POST) 76](#_Toc428525461)

[5.1.8 GetLiveCasinoBetHistory (GET) 77](#_Toc428525462)

[5.1.9 GetCtxmBetHistory (GET) 78](#_Toc428525463)

[5.1.10 Get3DSlotBetHistory (GET) 79](#_Toc428525464)

[5.2 Gold Deluxe (GD) – 컨펌 대기중 81](#_Toc428525465)

[5.2.1 CreateMember 82](#_Toc428525466)

[5.2.2 ValidateToken ( GET ) - TBD 82](#_Toc428525467)

[5.2.3 GetMemberBalance 82](#_Toc428525468)

[5.2.4 Credit 83](#_Toc428525469)

[5.2.5 Debit 84](#_Toc428525470)

[5.2.6 Cancel 85](#_Toc428525471)

[5.2.7 GetJackpotPool 87](#_Toc428525472)

[5.2.8 PlayGame 88](#_Toc428525473)

[5.2.9 GetBetHistory 88](#_Toc428525474)

[5.2.10 GetRNGBetHistory 90](#_Toc428525475)

[5.3 Microgaming 93](#_Toc428525476)

[5.3.1 LaunchGameService (POST) 93](#_Toc428525477)

[5.3.2 PlayerDetail (POST) 95](#_Toc428525478)

[5.3.3 CreateMember 96](#_Toc428525479)

[5.3.4 PlayerBalance (POST) 97](#_Toc428525480)

[5.3.5 Transaction (POST) 98](#_Toc428525481)

[5.3.6 Replay (what for?) 100](#_Toc428525482)

[5.4 Ezuigi (EZ) – 컨펌 대기중 102](#_Toc428525483)

[5.4.1 PlayGame (POST) – Open Game Lobby 103](#_Toc428525484)

[5.4.2 Authentication (POST) - Authentication of User API 103](#_Toc428525485)

[5.4.3 Debit (POST) 105](#_Toc428525486)

[5.4.4 Rollback (POST) 107](#_Toc428525487)

[5.4.5 Credit (POST) 109](#_Toc428525488)

[5.4.6 Error Codes 112](#_Toc428525489)

[5.5 XIN Gaming (XIN) 113](#_Toc428525490)

[5.5.1 Launching a Game 114](#_Toc428525491)

[5.5.2 Authentice (POST) 115](#_Toc428525492)

[5.5.3 Debit (POST) 116](#_Toc428525493)

[5.5.4 Credit (POST) 117](#_Toc428525494)

[5.5.5 Balance (POST) 118](#_Toc428525495)

[5.5.6 Refund (POST) 118](#_Toc428525496)

[5.5.7 Error Handling 120](#_Toc428525497)

[5.6 Top Trend Gaming (TTGS or Amaya) 120](#_Toc428525498)

[5.6.1 Login 121](#_Toc428525499)

[5.6.2 GameLaunch 124](#_Toc428525500)

[5.6.3 Logout 125](#_Toc428525501)

[5.6.4 Exist 126](#_Toc428525502)

[5.6.5 GetAccountNames 127](#_Toc428525503)

[5.6.6 UpdatePlayer 128](#_Toc428525504)

[5.6.7 Account Balance Detail (Unused) 129](#_Toc428525505)

[5.6.8 GetPlayerInformation 129](#_Toc428525506)

[5.6.9 Coded Bonus (Unused) 131](#_Toc428525507)

[5.6.10 GetBalance (From TTG to IMS) 131](#_Toc428525508)

[5.6.11 FundTransfer (From TTG to IMS) 132](#_Toc428525509)

[5.6.12 FundTransfer – Multiple Transactions (From TTG to IMS) 133](#_Toc428525510)

[5.6.13 FundTransfer – CombinedTransactions (From TTG to IMS) 135](#_Toc428525511)

[5.6.14 FundTransfer – Net Zero Transactions (From TTG to IMS) 137](#_Toc428525512)

[5.7 XPro Gaming (XPRO) 140](#_Toc428525513)

[5.7.1 CreateAccount 141](#_Toc428525514)

[5.7.2 ChangeAccount 142](#_Toc428525515)

[5.7.3 KickPlayerFromGames 143](#_Toc428525516)

[5.7.4 ChangeAccountStatus 144](#_Toc428525517)

[5.7.5 ChangeAccountPassword 145](#_Toc428525518)

[5.7.6 GetGamesListWithLimits 146](#_Toc428525519)

[5.7.7 RegisterToken 152](#_Toc428525520)

[5.7.8 GetExternalGamesList, GetExternalGameURL (Unused) 153](#_Toc428525521)

[5.7.9 GetLastWinners 153](#_Toc428525522)

[5.7.10 GetOperatorActivityReport (Unused) 155](#_Toc428525523)

[5.7.11 GetPlayersRejectedBets 155](#_Toc428525524)

[5.7.12 GetPlayersChatReport (Unused) 155](#_Toc428525525)

[5.7.13 GetPlayersRevenue 155](#_Toc428525526)

[5.7.14 PlayerGetBalance (POST) - From XPro to IMS 155](#_Toc428525527)

[5.7.15 PlayerGetBalances (POST) - From XPro to IMS 156](#_Toc428525528)

[5.7.16 Debit (POST) - From XPro to IMS 157](#_Toc428525529)

[5.7.17 Credit (POST) - From XPro to IMS 159](#_Toc428525530)

[5.7.18 CancelTransaction (POST) - From XPro to IMS 162](#_Toc428525531)

[5.7.19 CancelRound (POST) - From XPro to IMS 163](#_Toc428525532)

[5.7.20 GetBetHistory 164](#_Toc428525533)

[5.8 Laxino Gaming (Laxino) – TBD 176](#_Toc428525534)

[5.9 CMTech (CMTech) 176](#_Toc428525535)

[5.9.1 ValidateToken (GET) 176](#_Toc428525536)

[5.10 XTD (XTD) – provide SingleWallet API 177](#_Toc428525537)

[5.11 Asia Gaming (AG) 177](#_Toc428525538)

[5.11.1 CreatePlayerTicket (Post) – From IMS to AG 179](#_Toc428525539)

[5.11.2 ForwardGame (Form POST) 180](#_Toc428525540)

[5.11.3 GetBalance (GET) 182](#_Toc428525541)

[5.11.4 PostTransfer (GET) 183](#_Toc428525542)

[6 WalletToWallet API 185](#_Toc428525543)

[6.1 Winning FT (WFT) 185](#_Toc428525544)

[6.1.1 CreateMember (GET) 186](#_Toc428525545)

[6.1.2 Update (GET) 186](#_Toc428525546)

[6.1.3 CheckBalance (GET) 188](#_Toc428525547)

[6.1.4 Deposit 188](#_Toc428525548)

[6.1.5 Withdraw (GET) 189](#_Toc428525549)

[6.1.6 Login (GET) 190](#_Toc428525550)

[6.1.7 Logout (GET) 191](#_Toc428525551)

[6.1.8 CheckPayment (GET) 192](#_Toc428525552)

[6.1.9 CheckOnlineUsers (GET) 193](#_Toc428525553)

[6.1.10 Ticket (GET) 193](#_Toc428525554)

[6.1.11 Parlay (GET) 194](#_Toc428525555)

[6.1.12 Team (GET) 195](#_Toc428525556)

[6.1.13 League (GET) 196](#_Toc428525557)

[6.1.14 Fetch (GET) 197](#_Toc428525558)

[6.1.15 Mark Fetched (GET) 199](#_Toc428525559)

[6.1.16 GetJackpots (GET) 200](#_Toc428525560)

[6.2 Asia Sports Club (ASC) 201](#_Toc428525561)

[6.2.1 Register 202](#_Toc428525562)

[6.2.2 RegisterByLimit 203](#_Toc428525563)

[6.2.3 Login 205](#_Toc428525564)

[6.2.4 Login Member Site 206](#_Toc428525565)

[6.2.5 Logout 207](#_Toc428525566)

[6.2.6 GetOnlineMember 208](#_Toc428525567)

[6.2.7 Transfer 209](#_Toc428525568)

[6.2.8 CheckTransfer 210](#_Toc428525569)

[6.2.9 CheckTransferByAccount 211](#_Toc428525570)

[6.2.10 GetBalance 212](#_Toc428525571)

[6.2.11 UpdateMemberLimit 213](#_Toc428525572)

[6.2.12 GetMemberLimit 214](#_Toc428525573)

[6.2.13 GetBetSheet 215](#_Toc428525574)

[6.2.14 GetBetSheetByReport 217](#_Toc428525575)

[6.2.15 GetParlayBetSheet 220](#_Toc428525576)

[6.2.16 GetLeagueName 221](#_Toc428525577)

[6.2.17 GetTeamName 223](#_Toc428525578)

[6.2.18 GetMatchResult 225](#_Toc428525579)

[6.2.19 UpdateGroupComm 226](#_Toc428525580)

[6.2.20 UpdateStatus 227](#_Toc428525581)

[6.2.21 GetStatus 228](#_Toc428525582)

[6.3 M8 (M8) 230](#_Toc428525583)

[6.3.1 CreateMember (GET) 231](#_Toc428525584)

[6.3.2 Update (GET) 231](#_Toc428525585)

[6.3.3 CheckBalance (GET) 233](#_Toc428525586)

[6.3.4 Deposit 234](#_Toc428525587)

[6.3.5 Withdraw (GET) 234](#_Toc428525588)

[6.3.6 Login (GET) 235](#_Toc428525589)

[6.3.7 Logout (GET) 236](#_Toc428525590)

[6.3.8 CheckPayment (GET) 237](#_Toc428525591)

[6.3.9 CheckOnlineUsers (GET) 238](#_Toc428525592)

[6.3.10 Ticket (GET) 239](#_Toc428525593)

[6.3.11 Parlay (GET) 240](#_Toc428525594)

[6.3.12 Team (GET) 241](#_Toc428525595)

[6.3.13 League (GET) 242](#_Toc428525596)

[6.3.14 Fetch (GET) 243](#_Toc428525597)

[6.3.15 Mark Fetched (GET) 244](#_Toc428525598)

[6.4 XTD (XTD) – moved to SingleWallet API 245](#_Toc428525599)

[6.4.1 CreateMember 245](#_Toc428525600)

[6.4.2 GetMemberBalance 246](#_Toc428525601)

[6.4.3 Credit 246](#_Toc428525602)

[6.4.4 Debit 247](#_Toc428525603)

[6.4.5 PlayGame 248](#_Toc428525604)

[6.4.6 GetJackpotPool (Unused) 248](#_Toc428525605)

[6.5 SBO (SBO) 249](#_Toc428525606)

[6.5.1 RegisterPlayer (GET) 249](#_Toc428525607)

[6.5.2 Login(GET) 250](#_Toc428525608)

[6.5.3 UpdatePlayerStatus(GET) 251](#_Toc428525609)

[6.5.4 Deposit(GET) 252](#_Toc428525610)

[6.5.5 Withdraw (GET) 253](#_Toc428525611)

[6.5.6 CheckTransactionStatus (GET) 254](#_Toc428525612)

[6.5.7 GetPlayerBalance (GET) 255](#_Toc428525613)

[6.5.8 Logout (GET) 256](#_Toc428525614)

[6.5.9 Ping (GET) 257](#_Toc428525615)

[6.5.10 CheckStatus (GET) 258](#_Toc428525616)

[6.5.11 GetCustomerReport (GET) 259](#_Toc428525617)

[6.5.12 GetCustomerBetList(GET) 261](#_Toc428525618)

[6.6 IBC (IBC) 264](#_Toc428525619)

[6.6.1 CreateMember (GET) 264](#_Toc428525620)

[6.6.2 Login (GET) 265](#_Toc428525621)

[6.6.3 FundTransfer (GET) 266](#_Toc428525622)

[6.6.4 CheckUserBalance (GET) 267](#_Toc428525623)

[6.6.5 CheckFundTransfer (GET) 268](#_Toc428525624)

[7 Platform 269](#_Toc428525625)

[8 Appendix 269](#_Toc428525626)

[8.1 Terms Clarification 269](#_Toc428525627)

[8.1.1 API 269](#_Toc428525628)

[8.1.2 Business Terms 269](#_Toc428525629)

[8.1.3 Catogory ID Description 270](#_Toc428525630)

[8.1.4 agAccountTransaction vs gspBetTransaction 270](#_Toc428525631)

[8.1.5 Wallet Clarification 271](#_Toc428525632)

[8.1.6 Sports Bet Type (WFT) 271](#_Toc428525633)

[8.1.7 Sports Bet Limit (ASC) 272](#_Toc428525634)

[8.2 Transaction ID Management 272](#_Toc428525635)

[8.3 Language Code Management 272](#_Toc428525636)

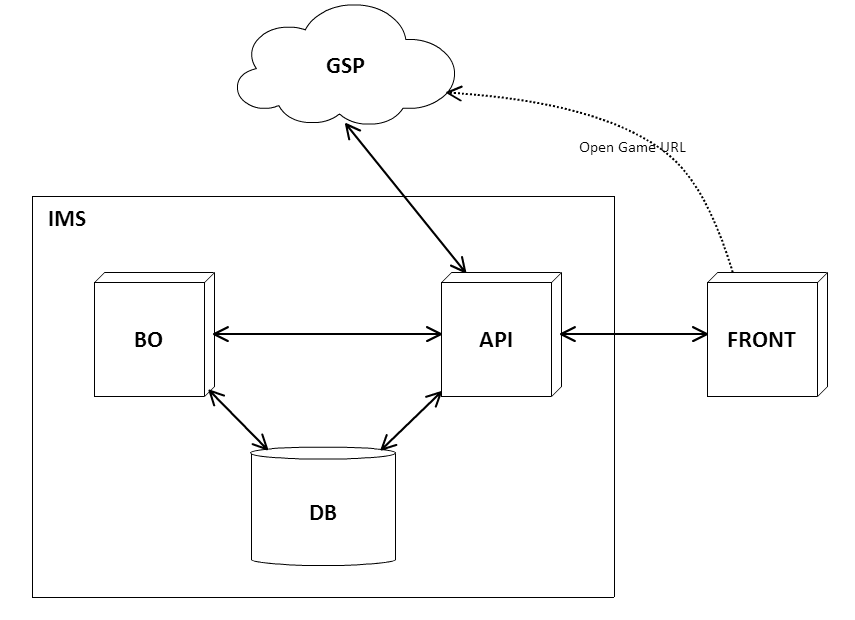
[8.4 Currency Code Management 273](#_Toc428525637)

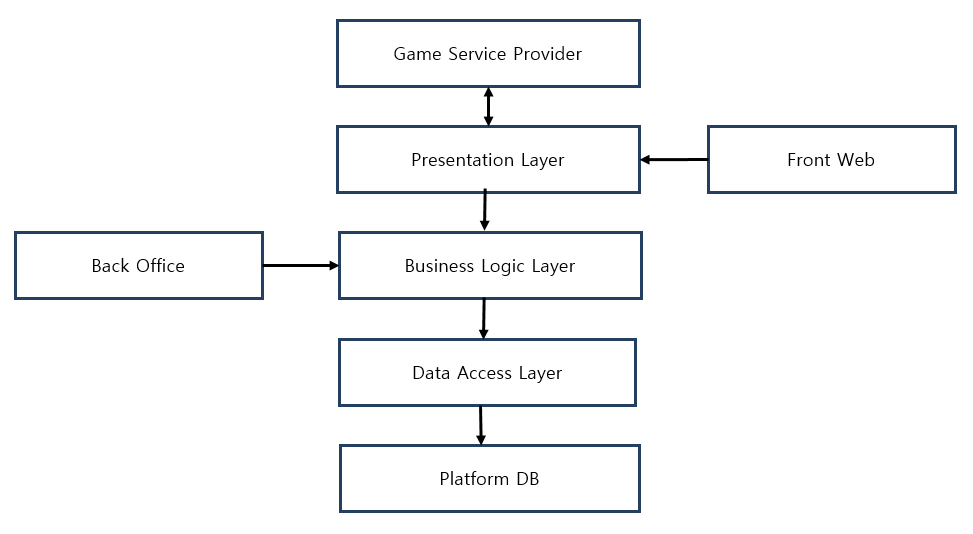
[8.5 Country Code Management 273](#_Toc428525638)

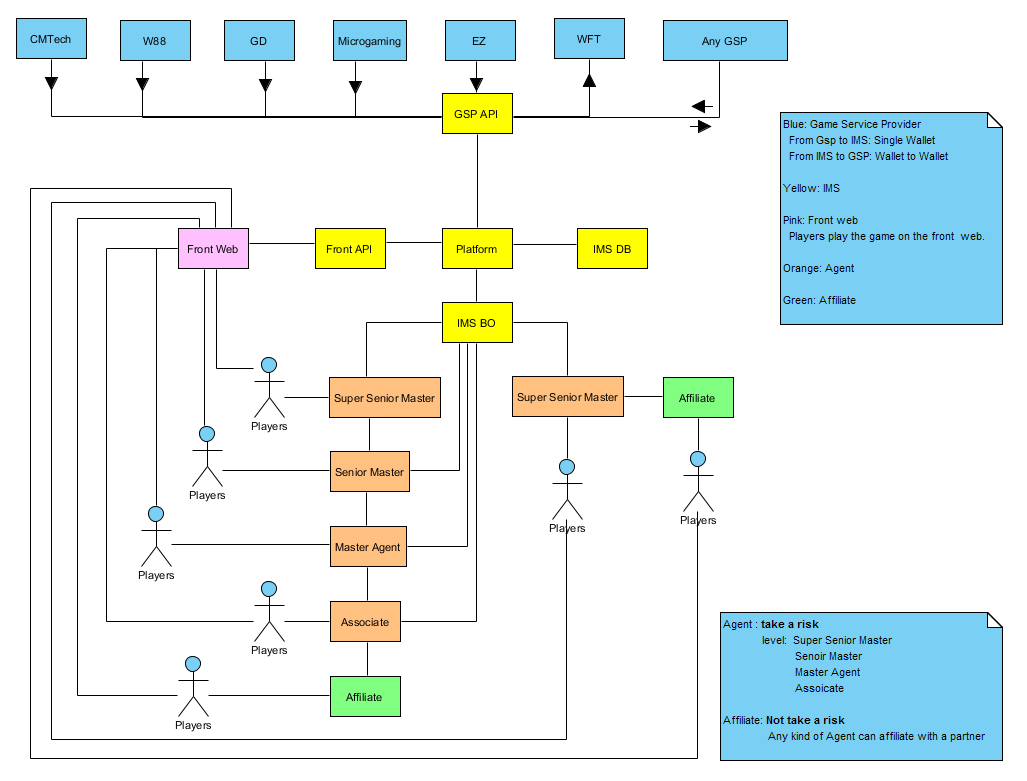
# **Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Updated By** | **Comments** |
| 1.0 | 2015-04-24 | Sunghwan Yoon | First Draft |
| 2.0 | 2015-04-30 | Sunghwan Yoon | API’s are distinguished by Single Wallet or Wallet to Wallet |
| 2.1 | 2015-05-04 | Sunghwan Yoon | Add TODO list to Front API |
| 2.1 | 2015-05-05 | Sunghwan Yoon | Add TODO list to Front API |
| 2.2 | 2015-05-06 | Sunghwan Yoon | First Draft of Front API |
| 2.3 | 2015-05-07 | Sunghwan Yoon | add SingleWallet API |
| 2.3 | 2015-05-27 | Sunghwan Yoon | add WalletToWallet API |
| 2.4 | 2015-05-28 | Sunghwan Yoon | modify with developing front API |
| 2.4 | 2015-06-01 | Sunghwan Yoon | apply an updated document for TTGS |
| 2.4 | 2015-06-12 | Sunghwan Yoon | add Player,SystemSetting, Operation, Agent, Fraud in Front API |
| 2.5 | 2015-06-16 | Sunghwan Yoon | add CheckUrl, CheckServerIp in Front API |
| 2.5 | 2015-06-30 | Sunghwan Yoon | add Finance API |
| 2.5 | 2015-07-01 | Sunghwan Yoon | add Maketing API |
| 2.6 | 2015-07-14 | Sunghwan Yoon | add SBO API |
| 2.6 | 2015-07-25 | Sunghwan Yoon | add GetAnnounceContents to Front API |
| 2.6 | 2015-08-05 | Sunghwan Yoon | Add IBC API  modify LoginToGsp Response |
| 2.6 | 2015-08-17 | Sunghwan Yoon | remove GetAgentGspGameList  add GetAgentProductGspList  add GetGspGameList |
| 2.7 | 2015-08-18 | SooYoung Yang | Edit WFT, M8 API |
| 2.7 | 2015-08-24 | Sunghwan Yoon | add SendFrontWebSMS |
| 2.8 | 2015-08-27 | SooYoung Yang | Edit style and add index |
| 2.9 | 2015-09-03 | Sunghwan Yoon | add clientNo in the API |
| 2.9 | 2015-09-24 | Sunghwan Yoon | add City, HomeTel, WorkTel, AddrState in Signup, GetPlayerDetail, EditPlayerDetail |
| 2.9 | 2015-09-30 | Sunghwan Yoon | add Credit Card Number in the PSPDepositResult API |
| 2.9 | 2015-10-21 | Sunghwan Yoon | add languageCd in the LoginToGsp API |
| 2.9 | 2015-10-27 | Sunghwan Yoon | Add XTD API  add creditCardNo, agBankAcctSeqNo, agBankNo in RequestDesposit API  add GetAgentBankAccount API |
| 2.10 | 2015-10-30 | Sunghwan Yoon | add accessToken in GetAgentGspList API and GetAgentProductGspList API |
| 2.10 | 2015-11-05 | Sunghwan Yoon | add agentNo, agentId, agentPw in the response of Login API |
| 2.10 | 2015-11-12 | Sunghwan Yoon | add bTag in the request and response of Login API |
| 2.10 | 2015-11-13 | Sunghwan Yoon | add MobileLogin API |
| 2.10 | 2015-11-17 | Sunghwan Yoon | add clientLocalTime and screen info in MobileLogin API  change bTag to agentBTag  add currencyBTag in the response of  GetAgentProductGspList API |

# **High Level Architecture**







# **TODO List**

|  |  |  |  |
| --- | --- | --- | --- |
| API 구분 | 기능 | 상세 기능 | 비고 |
| Front API | Board |  |  |
|  |  | 공지사항 | 3.4.1. GetAnnouncement |
|  |  | 이벤트 | 3.4.1. GetAnnouncement |
|  |  | FAQ | 3.4.1. GetAnnouncement |
|  |  | 1대1 문의 | 3.4.3. GetContactList  3.4.4. SetContact |
|  |  | Partner 문의 | 3.4.4. SetContact |
|  | GetDepositWeeklyList |  | 3.2.8. GetBankTransactionHistory |
|  | GetWithdrawWeeklyList |  | 3.2.8. GetBankTransactionHistory |
|  | GetFakeTransactionList |  | 3.2.8. GetBankTransactionHistory |
|  | GetAgentInfo |  | Agent telephone  Agent SNS id  Agent Email |
|  | GetBlackList |  | 3.7.1. CheckBlackList |
|  | PlayGames |  | 3.1.11. LoginToGsp |
|  | Bonus |  |  |
|  |  | GetCouponList | 3.5.1. GetPlayerCouponHistory |
|  |  | UseCoupon | 3.5.2. UseCoupon |
|  | GspList (id, name) |  | 3.6.2. GetSupportedGspList |
|  | GetBankCode |  | 3.3.7. GetPlayerBankCode |
|  | payment code number |  | 3.3.8. GetPspCode |
|  | deposit |  | 3.2.1 RequestDeposit  3.2.1 PSPDepositResult |
|  | withdraw |  | 3.2.2 RequestWithdrawal |
|  | PSP |  | reseller agreement  = PSP Fee: psp 에 따라 agent 에 부여하는 fee |
|  | 쪽지기능 |  | 3.4.1. GetAnnouncement에서 player 마다 검색하면 됨. |
|  | Login() |  | agent URL |
|  | Signup() |  | 가입한 agent url |
| BackOffice | SetFakeTransaction |  | Deposit  Withdrawal |
|  | SetCoupon |  |  |
|  | Jackpot |  | GSP Jackpot, progressive share: GSP에 지불해야함. |
|  | Fun money |  |  |
|  | agent balance |  | moneyTransfer (deposit)  : player balance(+)  agent balance(-):  moneyTransfer(withdrawal)  : player balance(-)  agent balance(+)  need a standard currency  agent balance history  useCoupon  : player balance(+)  agent balance(-)  financial adjustment  : player balance(+)  agent balance(-)  agent balance manual adjustment |
|  | AddCommentToContact |  |  |
|  | SetAnnouncement |  |  |
|  | CheckServer |  | API 호출 전, 서버/도메인/IP 확인후, 호출 여부 결정 |
|  | settlement report time |  | settlement gsp 기준으로 고정?  progress share? |
|  | bannedWords |  | 금지언어 |
|  | icon url management |  |  |
| GSP API |  |  |  |
|  | transfer |  | progressive share 필드 추가해야함. |
|  | progressive wallet |  | GSP API 문서 확인해야함.  GSP Balance? |
|  | Bet limit |  | Micro gaming API 체크 |
|  | 7월 15일 서버 장비 |  |  |
|  | SMS 연동 작업 |  |  |
| Poker API | gsp 별 game list + 아이콘 URL |  |  |
|  | AP별 기능별 각 메뉴 URL:  url type: internal url , external url  ML 참조 |  |  |
|  | SSO login? |  |  |

# **Front API**

This document describes the specification of network communication between Front Web and IMS API.

**Content-Type: application/json**

Header information

|  |  |
| --- | --- |
| **Header** | **VALUE** |
| UserIP | player ip address  it will be used when Login() |
| ServerIp | server ip address  Whenever Front API invoked, server ip address should be checked whether it is set in System Setting – Front Web or not. |
| UserAgent | User-Agent |
| VisiterURL | URL  Whenever Front API invoked, URL should be checked whether it is defined in agent URL management or not |
| agentId | agent id |
| agentPw | agent password |

* 1. Player
     1. Login ( POST )

**Mount Point:** /FrontAPI/Player.svc/login

The API is used to login to the platform system.

Call FAS SP to receive below information. (TBD)

SP Parameter should be included when calling Platform SP.

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATATYPE** | **DESCRIPTION** |
| DeviceNo | mandatory | int | device number |

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| nickname | mandatory | string | player nickname |
| password | mandatory | string | player password (SHA256 hashed password from Front) |
| clientLocalTime | mandatory | string | player local time (NOT UTC)  ISO\_8601: 1981-02-22T09:00:00 |
| loginChannelType | mandatory | Byte | login channel type  0 Unknown  1 Windows  2 Mac  3 Web  4 Mobile |
| agentUrlSeqNo | optional | int | agent url sequence number  : API will handle this value after CheckURL(). |
| screenWidth | mandatory | smallint | screen width size |
| screenHeight | mandatory | smallint | screen height size |
| accessToken | optional | string | access token   1. A player login from download poker client 2. The player try to use cashier system. 3. Poker client pass the accessToken to Front web. 4. Front web use SSO login by the access token. |
| agentBTag | optional | smallint | agent BTag |

Example

|  |
| --- |
| {  “nickname”:”syoon7”,  “password”:”syoon7”,  “clientLocalTime”:”1981-02-22T09:00:00”,  “loginChannelType”:1,  “screenWidth”:32767,  “screenHeight”:32767,  } |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| needToResetPwYn | mandatory | boolean | need to reset password |
| playerId | mandatory | string | player Id |
| accessToken | mandatory | string | access token |
| agentNo | mandatory | int | agent number |
| nickName | mandatory | string | player nickname |
| deviceNo | optional | Int64 | device number |
| AgentLoyaltyNo | mandatory | int | agent loyalty number |
| loyaltyLevel | mandatory | Byte | loyalty level to be used displayed loyalty image |
| currencyInfo | mandatory | CurrencyInfo | player currency info |
| languageNo | mandatory | int | language number |
| countryNo | mandatory | int | country number  retrieved from Maxmind DB. |
| ispName | optional | string | internet service provider name  retrieved from Maxmind DB. |
| city | mandatory | string | city name  retrieved from Maxmind DB. |
| needToEditPlayerYn | mandatory | Boolean | true: EditPlayer should be invoked |
| gspList | mandatory | List<GspInfo> | gsp list the player can play |
| agentId | mandatory | string | agent Id |
| agentPw | mandatory | string | agent password |
| agentBTag | mandatory | int | agentBTag  0 means that agent doesn’t select the bTag value |

Currency Code

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyNo | mandatory | int | currency number |
| currencyIsoCd | mandatory | string | currency ISO code |

Gsp Info

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gspNo | mandatory | int | gsp number |
| gspName | mandatory | string | gsp name |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| {  “needToResetPwYn”: false,  “playerId”: “CO\_7704”,  “accessToken”: “C927BBF4D79141D9B086ED952D243571”,  “agentNo”: 80000,  “nickname”: “test100”,  “deviceNo”: 0,  “agentLoyaltyNo”: null,  “loyaltyLevel”: null,  “loyaltyLevelNm”: null,  “currencyInfo”: {  “currencyNo”: 100,  “currencyIsoCd”: “USD”  },  “languageNo”: 100,  “countryNo”: null,  “ispName”: “”,  “city”: “”,  “needToEditPlayerYn”: true,  “gspList”: [  {  “gspNo”: 208,  “gspName”: “SBO”  }  ],  “agentBTag”: 1,  } |

* + 1. Logout ( PUT )

**Mount Point:** ~/FrontAPI/Player.svc/logout/{**accessToken**}/{**logoutTypeCd**}

The API is to be called when a player logs-out from Front Web.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |
| logoutTypeCd | mandatory | Byte | logout type code  1 Normal  2 Abnormal  3 Duplicate  duplicate: not allow multiple-login so that IMS make the previous accessToken expired |

Response

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK | idempotent |

* + 1. CheckDuplicateData ( POST )

**Mount Point:** /FrontAPI/Player.svc/checkDuplicate

The API is used to verify duplicated data during signup process.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| infoType | mandatory | Int | type  1: player nickname (should be unique per agent)  2: player email (Default: uncheck)  3: player telephone (Default: uncheck) |
| infoValue | mandatory | string | nickname or email or tel |

Example

|  |
| --- |
| {  “infoType”:1,  “infoValue”:”Test5”  } |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| isDuplicate | mandatory | Bool | duplicated data: true , otherwise false |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |

Example

|  |
| --- |
| {  “isDuplicate”: true  } |

* + 1. Signup ( POST )

**Mount Point:** /FrontAPI/Player.svc/signup

The country code will be automatically chosen based on the player’s IP address but it is configurable.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| affiliateId | optional | string | affiliate id |
| nickname | mandatory | string | player nickname used when Login() |
| password | mandatory | string | player password  (SHA256 hashed password from Front) |
| playerName | optional | string | player name |
| firstName | optional | string | player first name  scalability to respond to any customer’s needs  If exsit, PlayerName: firstName + lastName handled by SP |
| lastName | optional | string | player last name  scalability to respond to any customer’s needs  If exsit, playerName: firstName + lastName handled by SP |
| dateOfBirth | optional | string | birthday  ISO\_8601: 1981-02-22T09:00:00 |
| email | optional | string | email address |
| countryNo | optional | Int32 | country number |
| clientIp | optional | string | Signup Client IP address  : API can collect the IP address from the Header |
| phone | optional | string | cellular phone number |
| homeTel | optional | string | home telephone number |
| workTel | optional | string | work telephone number |
| smsYn | optional | Bool | enable to get SMS: true , otherwise: false |
| address | optional | string | address |
| addressDetail | optional | string | address detail |
| addressState | optional | string | address state |
| city | optional | string | city |
| zipCode | optional | string | postal code |
| currencyNo | optional | int | currency number |
| languageNo | optional | string | language number |
| securityQuestionNo | optional | Int16 | security question number  ( the question list stored in DB) |
| securityAnswer | optional | string | security question answer |
| hint | optional | string | password hint |
| gender | optional | Byte | gender  1: male  2: female |
| referrerNickName | optional | string | referrer player’s nickname |
| signupChannelType | optional | string | signup channel type  0 Unknown  1 Windows  2 Mac  3 Web  4 Mobile |
| reqisterUrl | mandatory | string | player’s register url |

Example

|  |
| --- |
| {  “affiliateId”:””,  “nickname”:”syoon3”,  “password”:”syoon3”,  “playerName”:”player name”,  “firstName”:”first name”,  “lastName”:”last name”,  “dateOfBirth”:”1981-02-22T09:00:00+09:00”,  “email”:”syoon3@aaa.com”,  “countryNo”:100,  “phone”:”012-345-6789”,  “smsYn”:true,  “address”:”address”,  “zipCode”:”123-456”,  “currencyNo”:100,  “languageNo”:100,  “securityQuestionNo”:1,  “securityAnswer”:”security answer”,  “hint”:”hint”,  “gender”:1,  “referrerNickName”:”referrer nick name”,  “signupChannerType”:1  } |

#### Response

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

2. 1. 1. ExtendAccessToken ( PUT ) – unused

This API is not exposed to outer system. It is only called by other APIs.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agentId | mandatory | string | agent id |
| accessToken | mandatory | string | access token |

#### Response

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | 0 | OK |
| 400 | -37 | INVALID\_ACCESS\_TOKEN |

2. 1. 6. GetPlayerDetail ( GET )

**Mount Point:** ~/FrontAPI/Player.svc/token/{**accessToken}**/playerDetail

The API is to look up player’s account detail.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| affiIiateId | optional | string | affiIiate id |
| nickname | optional | string | player nickname |
| playerName | optional | string | player name |
| firstName | optional | string | first name |
| middleName | optional | string | middle name |
| lastName | optional | string | last name |
| dateOfBirth | optional | string | birthday  ISO\_8601: 1981-02-22T09:00:00 |
| email | optional | string | email |
| countryNo | optional | int | country number |
| phone | optional | string | cellular phone number |
| homeTel | optional | string | home telephone number |
| workTel | optional | string | work telephone number |
| smsYn | optional | Bool | enable to get SMS: true , otherwise: false |
| address | optional | string | address |
| addressDetail | optional | string | address detail |
| addressState | optional | string | address state |
| city | optional | string | city |
| zipCode | optional | string | postal code |
| currencyInfo | mandatory | CurrencyInfo | player currency info |
| languageNo | optional | int | language number |
| securityQuestionNo | optional | Int | security question number  ( the question list stored in DB) |
| securityAnswer | optional | string | security question answer |
| hint | optional | string | password hint |
| gender | optional | byte | player’s gender  1: male  2: female |

CurrencyInfo

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyNo | mandatory | int | currency number |
| currencyIsoCd | mandatory | string | currency ISO code |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |

Example

|  |
| --- |
| {  “agentNo”: 10000,  “affiliateId”: “”,  “nickname”: “syoon1”,  “playerName”: “TestPlayerNm”,  “firstName”: “”,  “lastName”: “”,  “dateOfBirth”: “1981-02-22T09:00:00+09:00”,  “email”: “syoon1@test.com”,  “countryNo”: 506,  “clientIp”: null,  “phone”: “123-456-789”,  “smsYn”: false,  “address”: “Test Addr”,  “zipCode”: “123-56”,  “currencyInfo”: {  “currencyNo”: 100,  “currencyIsoCd”: “USD”  },  “languageNo”: 100,  “securityQuestionNo”: 1,  “securityAnswer”: “TEST”,  “hint”: “TestHint”  } |

* + 1. ChangePassword ( PUT )

**Mount Point:** ~/FrontAPI/Player.svc/changePassword

This API is to change player’s password.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |
| Password | mandatory | string | current password |
| newPassword | mandatory | string | new password to be set |
| nickname | mandatory | string | nickname |

Example

|  |
| --- |
| {  “accessToken”:”CA0546E88C644B77B6119BC5317273DE”,  “password”:”syoon2”,  “newPassword”:”syoon1”,  “accessToken”:”test20”,  } |

#### Response

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

* + 1. EditPlayerDetail ( PUT )

**Mount Point:** /FrontAPI/Player.svc/editPlayerDetail/

This API is to edit player’s information.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |
| playerName | optional | string | player name |
| firstName | optional | string | first name |
| lastName | optional | string | last name |
| dateOfBirth | optional | string | birthday  ISO\_8601: 1981-02-22T09:00:00Z |
| email | optional | string | email |
| countryNo | optional | int | country number |
| phone | optional | string | cellular phone number |
| homeTel | optional | string | home telephone number |
| workTel | optional | string | work telephone number |
| smsYn | optional | Bool | enable to get SMS: true , otherwise: false |
| address | optional | string | address |
| addressDetail | optional | string | address detail |
| addressState | optional | string | address state |
| city | optional | string | city |
| zipCode | optional | string | postal code |
| languageNo | optional | int | language number |
| securityQuestionNo | optional | int16 | security question number  ( the question list stored in DB) |
| securityAnswer | optional | string | security question answer |
| hint | optional | string | password hint |
| gender | optional | Byte | player’s gender  1: male  2: female |
| referrerNickname | optional | string | referrer player’s nickname |
| clientIp | optional | string | Client IP address  : API can collect the IP address from the Header |

Example

|  |
| --- |
| {  “accessToken”:”6A6A70C69714480D9EF9384BD8B92C27”,  “playerName”:”newPlayerNm”,  “firstName”:””,  “lastName”:””,  “dateOfBirth”:”1981-02-22T09:00:00+09:00”,  “email”:”syoon1@test.com”,  “countryNo”:506,  “phone”:”012-3456-7890”,  “smsYn”:true,  “address”:”new address”,  “zipCode”:”new zipcode”,  “languageNo”:100,  “securityQuestionNo”:1,  “securityAnswer”:”new security answer”,  “hint”:”new hint”,  “gender”:1,  “referrerNickname”:”new referrer nickname”,  } |

#### Response

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

* + 1. ForgotPassword ( PUT )

**Mount Point:** /FrontAPI/Player.svc/forgotPassword

This API is to use when player forgot his own password.

After login by tempPassword, the player should change the password.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| nickname | mandatory | string | player nickname |
| securityQuestionNo | mandatory | int16 | security question number |
| securityAnswer | mandatory | string | security question answer |
| clientNo |  |  |  |

Example

|  |
| --- |
|  |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| tempPassword | mandatory | string | once valid password |

Example

|  |
| --- |
| {  “tempPassword”: “961969”  } |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

* + 1. ForgotNickname ( POST )

**Mount Point:** ~/FrontAPI/Player.svc/forgotNickname

This API is to use when player forgot his own nickname for the login.

Email is not the unique data, so there is no way to find one’s nickname exactly.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| email | mandatory | string | email |

Example

|  |
| --- |
| {  “email”:”syoon3@aaa.com”  } |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| nicknameList | mandatory | List<string> | nickname list of players who have same email |

Example

|  |
| --- |
| {  “nicknameList”: [  “syoon3”,  “syoon5”,  “syoon6”,  “toh1”,  “syoon7”  ]  } |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |

* + 1. LoginToGsp ( POST )

**Mount Point:** / FrontAPI/Player.svc/loginToGspGame

This API is to login to GSP game

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token  SBO, |
| gspNo | mandatory | int | gsp No  SBO, |
| childGspNo | optional | int | child gsp No  Gameplay |
| productNo | optional | int | productNo  mandatory: GamePlay |
| platform | optional | string | platform  : Web, Mobile, Mini  mandatory: GamePlay |
| gameId | optional | string | gameId  mandatory: Slots |
| isFunMode | optional | string | isFunMode  mandatory: Slots |
| languageCd | optional | string | Default: en-us |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameURL | mandatory | string | game url |
| gspToken | optional | string | IBC(209) : store the session token into cookie and name the variable [g] before accessing Sportsbook |

Example

|  |
| --- |
| {  “gameURL”: “http:///\\/sbtest.ib.gsoft88.net\\/Deposit\_ProcessLogin.aspx?lang=en”,  “gspToken”: “a7b730aa-0ce3-4a1c-a4b1-3ab8d4489039”  } |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |

* + 1. SSOLogin ( GET )

**Mount Point:** ~/FrontAPI/Player.svc/token/{**accessToken}**/ssoLogin

The API is to validate the token as used SSO Login.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| affiIiateId | optional | string | affiIiate id |
| nickname | optional | string | player nickname |
| playerName | optional | string | player name |
| firstName | optional | string | first name |
| middleName | optional | string | middle name |
| lastName | optional | string | last name |
| dateOfBirth | optional | string | birthday  ISO\_8601: 1981-02-22T09:00:00 |
| email | optional | string | email |
| countryNo | optional | int | country number |
| phone | optional | string | phone number |
| smsYn | optional | bool | enable to get SMS: true , otherwise: false |
| address | optional | string | address |
| zipCode | optional | string | postal code |
| currencyInfo | mandatory | CurrencyInfo | player currency info |
| languageNo | optional | int | language number |
| securityQuestionNo | optional | Int | security question number  ( the question list stored in DB) |
| securityAnswer | optional | string | security question answer |
| hint | optional | string | password hint |
| needToEditPlayerYn | mandatory | Boolean | true: EditPlayer should be invoked |
| gspList | mandatory | List<GspInfo> | gsp list the player can play |

CurrencyInfo

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyNo | mandatory | int | currency number |
| currencyIsoCd | mandatory | string | currency ISO code |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |

Example

|  |
| --- |
| {  “needToResetPwYn”: false,  “accessToken”: “5D6C7A15889D40CCAD3410F47EA3D19C”,  “agentNo”: 10000,  “nickname”: “Test5”,  “deviceNo”: 0,  “agentLoyaltyNo”: 0,  “loyaltyLevel”: 0,  “currencyInfo”: {  “currencyNo”: 100,  “currencyIsoCd”: “USD”  },  “languageNo”: 100,  “countryNo”: 0,  “ispName”: “”,  “city”: “”,  “gspList”: [ ]  } |

* + 1. GetGameHistory ( POST )

**Mount Point:** ~/FrontAPI/Player.svc/gameHistory

The API is to retrieve the player’s game history list.

This is used by Backoffice Web.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agentNo | mandatory | int | agent number |
| playerId | mandatory | string | player id |
| startDate | mandatory | string | start date |
| endDate | mandatory | string | end date |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| playerBetList | mandatory | List<BetDetail> | bet history List |
| turnover | mandatory | decimal | turnover |

BetDetail

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
|  | mandatory | int | currency number |
| currencyIsoCd | mandatory | string | currency ISO code |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |

Example

|  |
| --- |
| {  “needToResetPwYn”: false,  “accessToken”: “5D6C7A15889D40CCAD3410F47EA3D19C”,  “agentNo”: 10000,  “nickname”: “Test5”,  “deviceNo”: 0,  “agentLoyaltyNo”: 0,  “loyaltyLevel”: 0,  “currencyInfo”: {  “currencyNo”: 100,  “currencyIsoCd”: “USD”  },  “languageNo”: 100,  “countryNo”: 0,  “ispName”: “”,  “city”: “”,  “gspList”: [ ]  } |

* + 1. MobileLogin ( POST )

**Mount Point:** /FrontAPI/Player.svc/mobileLogin

The API is used to login to the platform system and get the lobby url from GSP.

Header information

|  |  |
| --- | --- |
| **Header** | **VALUE** |
| UserIP | player ip address  it will be used when Login() |

SP Parameter should be included when calling Platform SP.

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATATYPE** | **DESCRIPTION** |
| DeviceNo | mandatory | int | device number |

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| nickname | mandatory | string | player nickname |
| password | mandatory | string | player password (SHA256 hashed password from Front) |
| productNo | optional | int | product number  10 Live Casino  20 Slot  30 Sportsbook  40 Poker  50 Others  60 Keno  70 Lottery |
| agentBTag | optional | int | agent Btag  parameter in the response of Login API |
| gspNo | mandatory | int | gsp number |
| gameId | optional | int | optional MG: 6625 default |
| clientLocalTime | mandatory | string | player local time (NOT UTC)  ISO\_8601: 1981-02-22T09:00:00 |
| screenWidth | mandatory | smallint | screen width size |
| screenHeight | mandatory | smallint | screen height size |

Example

|  |
| --- |
| {  “nickname”:”zar10”,  “password”:”5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8”,  “productNo”:10,  “languageCd”:”th”,  “agentBTag”:1,  “gspNo”:104,  “gameId”:6625,  “clientLocalTime”: “1981-02-22T09:00:00”,  “screenWidth”:32767,  “screenHeight”:32767  } |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameURL | mandatory | string | game url |
| gspToken | mandatory | string | gspToken |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| {  “gameURL”: “https://m.fortunabase.com?token=LPS:3:72441448:77164463:306:@:ad01d870-8a0f-11e5-bb85-32f76897aab2”,  “gspToken”: “LPS:3:72441448:77164463:306:@:ad01d870-8a0f-11e5-bb85-32f76897aab2”  } |

* 1. Finance
     1. RequestDeposit (POST)

**Mount Point:** /FrontAPI/Finance.svc/requestDeposit

This API is to use before front web site transfer player’s deposit request to PSP.

The player’s balance should be updated after admin’s approval.

|  |  |  |
| --- | --- | --- |
| 절차 | 매뉴얼 | PSP |
| 플레이어 입금 신청 | 입금신청-> RequestDeposit API 호출 | 입금신청 -> RequestDeposit API 호출 |
|  |  | PSP 호출 |
|  | 관리자 확인 | PSP 결과 Success / Failed   * PSPDepositResult API 호출 |



#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |
| pspNo | optional | int | payment service provider number  if null, manual |
| currencyAmount | mandatory | CurrencyAmount | transaction currency & amount |
| bankNo | optional | int | bank number in system setting |
| phone | optional | string | player phone number  don’t need to match player’s phone number |
| bankHolder | optional | string | depositor’s account holder name |
| depositDate | optional | string | deposit date  ISO\_8601: 1981-02-22T09:00:00 |
| depositPlace | optional | string | place which depositor request deposit  In case of China, be obligatory to write the deposit place |
| depositType | optional | string | deposit method  ex) ATM, phone banking |
| memo | optional | string | memo |
| clientIp | optional | string | Client IP address  : API can collect the IP address from the Header |
| creditCardNo | optional | string | credit card info used when PSP deposit |
| agBankAcctSeqNo | optional | int | agent bank account sequence number |
| agBankNo | optional | int | agent bank number |

CurrencyAmount

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyNo | optional | string | currency number |
| amount | mandatory | Int64 | transaction amount  should be positive value |

Example

|  |
| --- |
| {  “accessToken”:”E4034A44830F41F19E29C13F4763D449”,  “pspNo”:1,  “currencyAmount”:{  “currencyNo”:100,  “amount”:1000  },  “bankNo”:1,  “phone”:”012-345-6789”,  “bankHolder”:”bank holder name”,  « depositPlace » : »deposit place »,  « depositType » : »ATM »,  “memo”:”memo”,  } |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| depositNo | mandatory | Int64 | unique value |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |

Example

|  |
| --- |
| {  “depositNo”: 40  } |

* + 1. PSPDepositResult (PUT)

**unt Point:** / FrontAPI/Finance.svc/pspDepositResult

This API is to use after Front Web get a result of deposit from PSP



#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |
| depositNo | mandatory | Int64 | a response parameter of RequestDeposit() |
| pspTransacntionNo | mandatory | string | transaction id from PSP |
| pspKey | mandatory | string | key from PSP |
| pspStatusCd | mandatory | int | payement status from PSP  60102: Success  60103: Failed |
| creditCardNo | optional | string | SHA256 Encrypted Credit Card Information |

Example

|  |
| --- |
| {  “accessToken”:”E4034A44830F41F19E29C13F4763D449”,  “depositNo”:40,  “pspTransacntionNo”:”pspTransacntionNo”,  “pspKey”:”pspKey”,  “PspStatusCd”:60102,  “creditCardNo”:”thb1\_credit\_card\_no\_test”  } |

#### Response

Response Parameters

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |

Example

|  |
| --- |
|  |

* + 1. RequestWithdrawal (POST)

**Mount Point:** /FrontAPI/Finance.svc/RequestWithdrawal

This API is to use when player request a withdrawal.

Before admin approve, the player’s balance should be updated immediately.

|  |  |
| --- | --- |
| Action Detail | Who call |
| RequestWithdrawal API | Front web |
| MoneyTransfer (single wallet withdrawal) | Platform API |
| Approved -> SMS + Email  Requested: a withdrawal request is submitted  Approved: admin approve a withdrawal request  Rejected: admin reject a withdrawal request   * moneyTransfer (transaction descripton )   Canceled: player cancel a withdrawal request | BO |



#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |
| currencyAmount | mandatory | CurrencyAmount | transaction currency & amount |
| bankNo | mandatory | int | bank number in system setting |
| phone | mandatory | string | player phone number  don’t need to match player’s phone number |
| bankHolder | mandatory | string | withdrawal’s bank account holder name |
| bankAccountNo | mandatory | string | bank account number |
| bankAccountType | optional | string | bank account type  In case of China, mandatory |
| bankPlace | optional | string | bank place which a player request a withdrawal  In case of China, mandatory |
| bankOffice | optional | string | bank office  In case of China, mandatory |
| memo | optional | string | memo |
| clientIp | optional | string | client IP address  : API can collect the IP address from the Header |

CurrencyAmount

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyNo | optional | string | currency number |
| amount | mandatory | decimal | transaction amount |

Example

|  |
| --- |
| {  “accessToken”:”2668E55B10CB4B39969CC7150DCF68B0”,  “currencyAmount”:{  “currencyNo”:100,  “amount”:1000  },  “bankNo”:1,  “phone”:”012-345-6789”,  “bankHolder”:”bank holder name”,  “bankAccountNo”:”012-34-567-89”,  “bankAccountType”:”bank account type”,  “bankPlace”:”bank place”,  “bankOffice”:”bank office”,  “memo”:”memo”,  } |

#### Response

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |

* + 1. WalletTransfer (POST)

**Mount Point:** / FrontAPI/Finance.svc/walletTransfer

This API is to transfer money between IMS single wallet and GSP wallet.



#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |
| currencyAmount | mandatory | CurrencyAmount | transaction currency & amount |
| gspNo | mandatory | int32 | gsp number |
| productNo | mandatory | int32 | product number  obtain by 3.3.9. GetAgentGspGameList  If GSP support a few product, any product number in the list will be fine.  It is used when specify a transaction class code. |
| fromImsToGsp | mandatory | bool | true: IMS -> GSP  false: GSP -> IMS |

CurrencyAmount

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyNo | optional | string | currency number |
| amount | mandatory | decimal | transaction amount |

Example

|  |
| --- |
| {  “accessToken”:”DB43F53589A546A89C6E48B6E390C2F9”,  “currencyAmount”:{  “currencyNo”:100,  “amount”:10000  },  “gspNo”:101,  “productNo”:40,  “fromImsToGsp”:false  } |

#### Response

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |

* + 1. GetBalance ( GET )

**Mount Point:** / FrontAPI/Finance.svc/token/{**accessToken**}/balance

This API is to look up a player’s balances and currencies in IMS and Every GSP.



#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| mainBalance | mandatory | CurrencyAmount | player’s balance in IMS single wallet |
| gspBalance | mandatory | Dictionary<string, CurrencyAmount > | player’s every wallet to wallet balances  list of <gspId, CurrencyAmount > |

CurrencyAmount

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyIsoCd | mandatory | string | currency ISO code |
| amount | mandatory | decimal | the latest player’s balance amount |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |

Example

|  |
| --- |
| {  “mainBalance”: {  “currencyIsoCd”: “USD”,  “amount”: 123.4567  },  “gspBalance”: [  {  “Key”: “W88 gsp code”,  “Value”: {  “currencyIsoCd”: “USD”,  “amount”: 23.456  }  },  {  “Key”: “GD gsp code”,  “Value”: {  “currencyIsoCd”: “USD”,  “amount”: 34.567  }  }  ]  } |

* + 1. GetMainBalance ( GET )

**Mount Point:** / FrontAPI/Finance.svc/token/{**accessToken**}/mainBalance

This API is to look up player’s balance and currency in IMS.

player’s accessToken extend the expiration time as the API invoked.



#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| singleBalance | mandatory | CurrencyAmount | player’s balance in IMS single wallet |

CurrencyAmount

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyIsoCd | mandatory | string | currency ISO code |
| currencyNo | mandatory | int | currency Number |
| amount | mandatory | decimal | the latest player’s balance amount |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |

Example

|  |
| --- |
| {  “singleBalance”: {  “currencyIsoCd”: “USD”,  “amount”: 123.4567  }  } |

* + 1. GetGspBalance( GET )

**Mount Point:** / FrontAPI/Finance.svc/token/{**accessToken**}/gspBalance

This API is to look up player’s balances and currencies in Every Game Service Provider.



#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gspBalance | mandatory | Dictionary<string, CurrencyAmount> | player’s wallet to wallet balance  list of <gspId, Balance> |

CurrencyAmount

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyIsoCd | optional | string | currency ISO code |
| amount | mandatory | decimal | the latest player’s balance amount |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |

Example

|  |
| --- |
| {  “gspBalance”: [  {  “Key”: “W88 gsp code”,  “Value”: {  “currencyIsoCd”: “USD”,  “amount”: 23.456  }  },  {  “Key”: “GD gsp code”,  “Value”: {  “currencyIsoCd”: “USD”,  “amount”: 34.567  }  }  ]  } |

* + 1. GetPlayerTransactionHistory ( POST )

**Mount Point:** / FrontAPI/Finance.svc/playerTransactionHistory

This API is to look up a player’s transaction history





#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |
| startDate | mandatory | string | start date UTC timezone  ISO\_8601: 1981-02-22T09:00:00 |
| endDate | mandatory | string | end date UTC timezone  ISO\_8601: 1981-02-22T09:00:00 |
| languageNo | optional | int | language number |
| page | optional | int | page number  default: 1 |
| pageRows | optional | int | page row count  default: 100 |
| isBonusOnly | optional | bool | true: bonus only  false: transfer, deposit, credit |

Example

|  |
| --- |
| {  “accessToken”:”String content”,  “startDate”:”1981-02-22T09:00:00”,  « endDate » : »1981-02-22T09 :00 :00”,  « LanguageNo » :100,  « page » :1,  “pageRows”:100  } |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| transactionList | mandatory | List<TransactionDetail> | List of player’s money transfer history |
| totalCount | mandatory | int | count of items in list |

Transaction Detail

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyAmount | mandatory | CurrencyAmount | transaction currency & amount |
| gspName | mandatory | string | gsp name |
| transactionId | mandatory | string | transaction id |
| transactionDate | mandatory | string | transaction date UTC timezone  ISO\_8601 : 1981-02-22T09 :00 :00 |
| transactionClass | mandatory | string | transaction class code: Poker, Casino and so on |
| transactionSource | mandatory | string | transaction source code  ex) EZ, W88 |
| transactionType | mandatory | string | transaction type code  ex) Deposit, withdrawal, Transfer |
| transactionDescription | mandatory | string | transaction description code  ex) Cash Transfer To GSP  Cash Transfer From GSP  Customer Deposit  Manual Deposit |

CurrencyAmount

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyIsoCd | optional | string | currency ISO code |
| amount | mandatory | numeric(19,2) | transaction amount |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

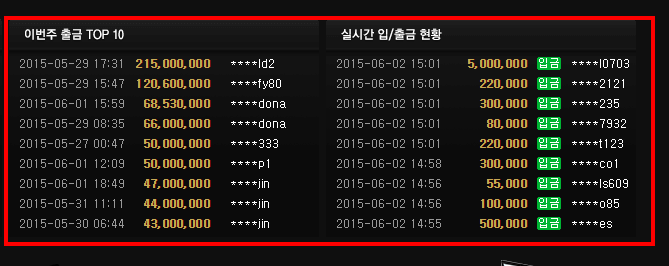
|  |
| --- |
| {  “transactionList”: [  {  “currencyAmount”: {  “currencyIsoCd”: “USD”,  “amount”: 10000  },  “transactionId”: “393”,  “transactionDate”: “2015-06-30T03:16:16”,  “transactionClass”: “Poker”,  “transactionSource”: “EZ”,  “GspName”: “”,  “transactionType”: “Transfer”,  “transactionDescription”: “Cash Transfer From GSP”  },  ]  } |

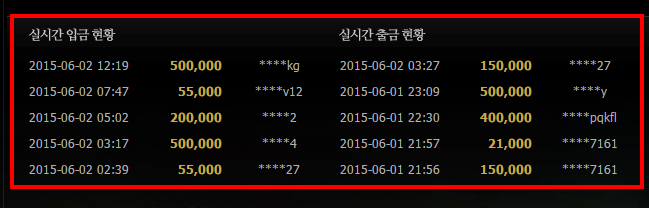
* + 1. GetPaymentTransactionHistory ( POST )

Mount Point: / FrontAPI/Finance.svc/paymentTransactionHistory

This API is to look up a list of top players’ bank transaction for last 7 days.

Agent 기준: Deposit/Withdrawal: 각 N개, Top – 10개







#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| topCount | mandatory | int | top count of a list |
| includeBotTransaction | optional | bool | default: false, otherwise: true |
| languageNo | optional | int | language number |

Example

|  |
| --- |
| {  “topCount”:255,  “includeBotTransaction”:true,  “languageNo”:100  } |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| topWithdrawalList | mandatory | List<transactionDetail> | top N withdrawal transactions  order by amount desc |
| currentDepositList | mandatory | List<transactionDetail> | current depsosit transactions  order by transaction date desc |
| currentWithdrawalList | mandatory | List<transactionDetail> | current withdrawal transactions  order by transaction date desc |

TransactionDetail

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| nickname | mandatory | string | player nickname |
| currencyAmount | mandatory | CurrencyAmount | currency ISO code and amount |
| transactioDate | mandatory | Datetime | transaction date  ISO\_8601: 1981-02-22T09:00:00 |

CurrencyAmount

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyIsoCd | optional | string | currency ISO code |
| amount | mandatory | numeric(19,2) | transaction amount |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| {  “topWithdrawalList”: [  {  “CurrencyAmount”: {  “currencyIsoCd”: “USD”,  “amount”: 1000  },  “Nickname”: “syoon9”,  “TransactionDate”: “2015-06-17T02:43:51”  }  ],  “currentDepositList”: [  {  “CurrencyAmount”: {  “currencyIsoCd”: “USD”,  “amount”: 1000  },  “Nickname”: “syoon9”,  “TransactionDate”: “2015-06-17T02:42:50”  }  ],  “currentWithdrawalList”: [  {  “CurrencyAmount”: {  “currencyIsoCd”: “USD”,  “amount”: 1000  },  “Nickname”: “syoon9”,  “TransactionDate”: “2015-06-17T02:43:51”  }  ]  } |

* + 1. GetGameTransactionHistory ( POST )

**Mount Point:** / FrontAPI/Finance.svc/gameTransactionHistory

This API is to look up a player’ game transaction history



#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |
| startDate | mandatory | string | start date UTC timezone  ISO\_8601: 1981-02-22T09:00:00 |
| endDate | mandatory | string | end date UTC timezone  ISO\_8601: 1981-02-22T09:00:00 |
| gspNo | optional | List<int> | gsp number |
| gspGameNo | optional | List<int> | gsp game number |
| languageNo | optional | int | language number |

Example

|  |
| --- |
| {  “accessToken”:”A9FCEC41E3E041FB8341C6CE76411FE4”,  “startDate”:”2015-01-01”,  “endDate”:”2015-12-31”,  “gspNo”:[301,102],  } |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| transactionList | mandatory | List | List of player’s transactionDetail |

Transaction Detail

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyAmount | mandatory | CurrencyAmount | transaction currency & amount |
| gspName | mandatory | string | gsp name |
| gspGameName | mandatory | string | gsp game name |
| transactionId | mandatory | string | transaction id |
| transactioDate | mandatory | string | transaction date (UTC timezone)  ISO\_8601 : 1981-02-22T09 :00 :00 |
| transactionClass | mandatory | string | transaction class: Poker, Casino |
| transactionSource | mandatory | string | transaction source |
| gspGameName | mandatory | string | gspGameName |
| transactionType | mandatory | string | transaction type |
| transactionDescription | mandatory | string | transaction description |

CurrencyAmount

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyIsoCd | optional | string | currency ISO code |
| amount | mandatory | numeric(19,2) | transaction amount |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| {  “transactionList”: [  {  “currencyAmount”: {  “currencyIsoCd”: “USD”,  “amount”: -10000  },  “transactionId”: “382”,  “transactionDate”: “2015-06-29T11:56:48”,  “transactionClass”: “Poker”,  “transactionSource”: “Cash Game Server”,  “GspName”: “CM Tech”,  “gspGameName”: “”,  “transactionType”: “Debit”,  “transactionDescription”: “Buy-In”  },  {  “currencyAmount”: {  “currencyIsoCd”: “USD”,  “amount”: -10000  },  “transactionId”: “381”,  “transactionDate”: “2015-06-29T11:56:39”,  “transactionClass”: “Poker”,  “transactionSource”: “Cash Game Server”,  “GspName”: “CM Tech”,  “gspGameName”: “”,  “transactionType”: “Debit”,  “transactionDescription”: “Buy-In”  }  ]  } |

* + 1. PSP Deposit / Withdraw => Robert
  1. SystemSetting
     1. GetSecurityQ**uestio**nList ( GET )

**Mount Point:** / FrontAPI/SystemSetting.svc/languageNo/{**languageNo**}/securityQuestions

This API is to look up Security Question List.

Multi Language information will be provided.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| languageNo | mandatory | string | language number (conver to int) |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| securityQstList | mandatory | List< SecurityQuestionInfo> | security question list |

SecurityQuestionInfo

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| questionNo | mandatory | Int16 | security question number |
| questionDescription | mandatory | string | security question description  default: English |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |

Example

|  |
| --- |
| {  “securityQstList”: [  {  “questionNo”: 1,  “questionDescription”: “Mother’s name”  },  {  « questionNo » : 2,  « questionDescription » : « Pet’s name »  },  {  « questionNo » : 3,  « questionDescription » : « Favorite food »  },  {  « questionNo » : 4,  « questionDescription » : « Favorite color »  },  {  « questionNo » : 5,  « questionDescription » : « Favorite actor »  },  {  « questionNo » : 6,  “questionDescription”: “Favorite athlete”  }  ]  } |

* + 1. GetLanguageList ( GET )

**Mount Point:** /FrontAPI/SystemSetting.svc/languages

This API is to look up a list of system supported languages.

Multi Language information will be provided.



#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| languageList | mandatory | list<LanguageInfo> | language info list |

LanguageInfo

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| languageNo | mandatory | Int32 | language number |
| languageName | mandatory | string | language name  default: English |
| lnIso639\_1 | mandatory | string | language ISO 639-1 |
| lnIso639\_2 | mandatory | string | language ISO 639-2 |
| lnIso639\_3 | mandatory | string | language ISO 639-3 |
| lnBrowserCd | mandatory | string | Browser language code |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |

Example

|  |
| --- |
| {  “languageList”: [  {  “languageNo”: 100,  “languageName”: “English”,  “lnIso639\_1”: “en”,  “lnIso639\_2”: “eng”,  “lnIso639\_3”: “eng”,  “lnBrowserCd”: “en-us”  }  ]  } |

* + 1. GetAgentCountryList ( GET )

**Mount Point:** /FrontAPI/SystemSetting.svc/agentCountries

This API is look up a list of Country Code based on header agent info.

Multi Language information will be provided.

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| countryList | mandatory | list<CountryInfo> | country list |

CountryInfo

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| countryNo | mandatory | int | country number |
| countryName | mandatory | string | country name  default: English |
| countryIso3166\_1\_A2 | mandatory | string | country ISO 3166 alpha-2 |
| countryIso3166\_1\_A3 | mandatory | string | country ISO 3166 alpha-3 |
| countryCallingCd | mandatory | string | country telephone code |

Example

|  |
| --- |
| {  “countryList”: [  {  “countryNo”: 100,  “countryName”: “United States”,  “countryIso3166\_1\_A2”: “US”,  “countryIso3166\_1\_A3”: “USA”,  “countryCallingCd”: 1  },  {  “countryNo”: 500,  “countryName”: “China”,  “countryIso3166\_1\_A2”: “CN”,  “countryIso3166\_1\_A3”: “CHN”,  “countryCallingCd”: 86  }  ]  } |

* + 1. GetAgentCurrencyList (GET)

**Mount Point:** /PlatformClientService/SystemSetting.svc/agentCurrencies

This API is to look up a list of supported currency per agent id in header.

Multi Language information will be provided.

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyList | mandatory | list<CurrencyInfo> | currency list |

CurrencyInfo

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyNo | mandatory | int | currency number |
| currencyName | mandatory | string | currency name  default: English |
| currencyIsoCd | mandatory | string | currency ISO code |

Example

|  |
| --- |
| {  “currencyList”: [  {  “currencyNo”: 100,  “currencyName”: “United States Dollar”,  “currencyIsoCd”: “USD”  },  {  “currencyNo”: 500,  “currencyName”: “China Yuan Renminbi”,  “currencyIsoCd”: “CNY”  }  ]  } |

* + 1. GetPlayerCurrencyCodeList ( GET ) – (Unused)

**Mount Point:** / FrontAPI/SystemSetting.svc/token/{**accessToken**}/currencies

This API is to look up a list of currency code that the current player is allowed to use.

It will not be used until IMS supports the player’s multi currency system.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agentId | mandatory | string | agent id |
| accessToken | mandatory | string | access token |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyList | mandatory | list | currency list |

CurrencyInfo

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| priority | mandatory | int | currency priority |
| currencyNo | mandatory | int | currency number |
| currencyIsoCd | mandatory | string | currency ISO code |

Example

|  |
| --- |
| [  {  “priority”: 1,  “currencyNo”: 1,  “currencyIsoCd”: “USD”  }  ] |

* + 1. GetPlayerBankList ( GET )

**Mount Point:** / FrontAPI/SystemSetting.svc/currencyNo/{**currencyNo**}/agentBankList

This API is to look up a list of bank code regarding to current player’s currency.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyNo | mandatory | Int32 | player currency number |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| bankList | mandatory | list<BankInfo> |  |

Bank Info

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| bankNo | mandatory | Int32 | bank no |
| bankName | mandatory | string | bank name written by agent admin |

Example

|  |
| --- |
| {  “bankList”: [ ]  } |

* + 1. GetAgentPspList ( GET )

**Mount Point:** / FrontAPI/SystemSetting.svc/currencyNo/{**currencyNo**}/agentPspList

This API is to look up a list of PSP per agent and currency.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyNo | mandatory | string | player currency number |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| pspList | mandatory | list<PspInfo> | payment service provider |

Psp Info

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| pspNo | mandatory | int | psp number |
| pspName | mandatory | string | psp name |

Example

|  |
| --- |
|  |

* + 1. GetAgentGspList ( GET )

**Mount Point:** /FrontAPI/SystemSetting.svc/agentGspList?token={**accessToken**}

This API is to retrieve a supported GSP list of agent in the Header.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | optional | string | access token |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| mainWalletList | mandatory | list<GspInfo> | main wallet game service provider list |
| gspWalletList | mandatory | list<GspInfo> | gsp wallet game service provider list |

GspInfo

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gspNo | mandatory | int | gsp number |
| gspName | mandatory | string | gsp name |
| gspCode | mandatory | string | gsp code |
| singleWalletYn | mandatory | bool | true: single wallet gsp, otherwise: false |

Example

|  |
| --- |
| {  “mainWalletList”: [  {  “gspNo”: 101,  “gspName”: “EZ”,  “gspId”: “EZ”,  “gspPw”: “EZ”,  “gspCode”: “EZ”,  “singleWalletYn”: true  },  {  “gspNo”: 102,  “gspName”: “GAMEPLAY”,  “gspId”: “GP”,  “gspPw”: “GP”,  “gspCode”: “GP”,  “singleWalletYn”: true  },  {  “gspNo”: 102,  “gspName”: “GAMEPLAY”,  “gspId”: “GP”,  “gspPw”: “GP”,  “gspCode”: “GP”,  “singleWalletYn”: true  },  {  “gspNo”: 103,  “gspName”: “Gold Deluxe”,  “gspId”: “GD”,  “gspPw”: “GD”,  “gspCode”: “GD”,  “singleWalletYn”: true  }  ],  “gspWalletList”: []  } |

* + 1. GetAgentProductGspList ( GET )

**Mount Point:** /FrontAPI/SystemSetting.svc/productGspList?token={accessToken}

This API is to retrieve a supported GSP game list of agent.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | optional | string | access token |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| productList | mandatory | list<GspProduct> | roduct list |

GspInfo

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| productNo | mandatory | int | product number  10 Live Casino  20 Slot  30 Sportsbook  40 Poker  50 Others |
| gspList | mandatory | list<GspInfo> | gsp list |

GspInfo

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gspNo | mandatory | int | gsp number |
| gspName | mandatory | string | gsp name |
| currencyBTag | mandatory | int | gsp currency BTag  0 means doesn’t exist for GSP currency BTag |

Example

|  |
| --- |
| {  "productList": [  {  "productNo": 10,  "gspList": [  {  "gspNo": 101,  "gspName": "EZ",  "currencyBTag": 0  },  {  "gspNo": 102,  "gspName": "GAMEPLAY",  "currencyBTag": 0  },  {  "gspNo": 104,  "gspName": "Microgaming",  "currencyBTag": 0  },  {  "gspNo": 112,  "gspName": "UC8",  "currencyBTag": 0  },  {  "gspNo": 204,  "gspName": "Asia Gaming",  "currencyBTag": 0  }  ]  },  {  "productNo": 20,  "gspList": [  {  "gspNo": 102,  "gspName": "GAMEPLAY",  "currencyBTag": 0  },  {  "gspNo": 104,  "gspName": "Microgaming",  "currencyBTag": 0  },  {  "gspNo": 106,  "gspName": "XIN Gaming",  "currencyBTag": 0  },  {  "gspNo": 109,  "gspName": "Inteplay",  "currencyBTag": 0  },  {  "gspNo": 112,  "gspName": "UC8",  "currencyBTag": 0  }  ]  },  {  "productNo": 30,  "gspList": [  {  "gspNo": 201,  "gspName": "Asia Sports Club",  "currencyBTag": 0  },  {  "gspNo": 202,  "gspName": "M8",  "currencyBTag": 0  },  {  "gspNo": 203,  "gspName": "Winning FT",  "currencyBTag": 0  },  {  "gspNo": 208,  "gspName": "SBO",  "currencyBTag": 0  },  {  "gspNo": 209,  "gspName": "IBC",  "currencyBTag": 0  }  ]  },  {  "productNo": 40,  "gspList": [  {  "gspNo": 301,  "gspName": "CM Tech",  "currencyBTag": 0  }  ]  },  {  "productNo": 50,  "gspList": [  {  "gspNo": 210,  "gspName": "CF69",  "currencyBTag": 0  },  {  "gspNo": 211,  "gspName": "Muay18",  "currencyBTag": 0  }  ]  }  ]  } |

* + 1. CheckURL( GET )

**This API is not exposed to outer system. It is only called by other APIs.**

This API is to check whether the domain is valid or not.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| visiterUrl | mandatory | string | visiterUrl in the header |

* + 1. CheckServerIP( GET )

**This API is not exposed to outer system. It is only called by other APIs.**

This API is is to check whether the IP is valid or not.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| serverIp | mandatory | string | server ip in the header |

* + 1. GetCountryInfoByIP( GET )

Mount Point: /FrontAPI/SystemSetting.svc/ip/{ip}/countryInfo

This API is to retrieve a country information from Maxmind.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| ip | mandatory | string | player’s ip address |

***Response***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| countryIso3166\_1\_A2 | mandatory | string | countryIso3166\_1\_Alpha 3 |
| countryCallingCd | mandatory | string | country Calling Code |

* + 1. **GetGspGameList**( POST )

Mount Point: /FrontAPI/SystemSetting.svc/gspGameList

This API is to retrieve a country information from Maxmind.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gspNo | mandatory | int | gsp number |
| childGspNo | optional | int | child gsp number |
| platform | mandatory | string | platform name  Web, Mobile, Mini |
| category | mandatory | string | game category  Arcades  Card\_Games  Keno  Mobile  Multihand\_Poker  Pyramid\_Poker  Slots  Slots\_3d  Soft\_Games  Table\_Games  Video\_Poker |
| androidYn | mandatory | bool | true: android OS device  false: default |
| iosYn | mandatory | bool | true: iOS device  false: default |
| isFunMode | optional | bool | true: fun mode  false: default |

***Response***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gspGameList | mandatory | List<GameInfo> | game info list |

GameInfo

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameId | mandatory | string | game id |
| gameName | mandatory | string | game name |

Example

|  |
| --- |
| {  "gspGameList": [  {  "gameId": "soccer",  "gameName": "2014 Soccer Slot"  },  {  "gameId": "baseball",  "gameName": "Baseball"  },  {  "gameId": "bikinibeach",  "gameName": "Bikini Beach"  },  {  "gameId": "boxing",  "gameName": "Boxing"  },  {  "gameId": "kpop",  "gameName": "K-POP"  },  {  "gameId": "threekingdoms",  "gameName": "Three Kingdoms"  },  {  "gameId": "underwaterworld",  "gameName": "Underwater World"  },  {  "gameId": "romanempire",  "gameName": "Roman Empire"  },  {  "gameId": "nightclub",  "gameName": "Night Club"  },  {  "gameId": "mafia",  "gameName": "Mafia"  },  {  "gameId": "monkeyking",  "gameName": "The Monkey King"  },  {  "gameId": "redchamber",  "gameName": "The Red Chamer"  },  {  "gameId": "wildwildwestern",  "gameName": "Wild Wild Western"  },  {  "gameId": "casinoroyale",  "gameName": "Casino Royale"  },  {  "gameId": "jacksorbetter",  "gameName": "Captain Jack or Better"  },  {  "gameId": "rouletteAmerican",  "gameName": "Roulette American"  },  {  "gameId": "rouletteEuropean",  "gameName": "Roulette European"  },  {  "gameId": "roulettePro",  "gameName": "Roulette Pro"  },  {  "gameId": "ladyluck",  "gameName": "Lady Luck"  },  {  "gameId": "blackjack",  "gameName": "Blackjack"  },  {  "gameId": "forbiddenchamber",  "gameName": "Forbidden Chamber"  },  {  "gameId": "littlemonsters",  "gameName": "Little Monsters"  },  {  "gameId": "fruitilicious",  "gameName": "Fruit Paradise"  },  {  "gameId": "freedomfighter",  "gameName": "Freedom Fighter"  },  {  "gameId": "ninetailedninja",  "gameName": "Nine Tailed Ninja"  },  {  "gameId": "trickortreat",  "gameName": "Trick Or Treat"  },  {  "gameId": "desertoasis",  "gameName": "Desert Oasis"  },  {  "gameId": "queenbee",  "gameName": "Queen Bee"  },  {  "gameId": "nutcracker\_ld",  "gameName": "The Nutcracker"  },  {  "gameId": "winterwonderland\_ld",  "gameName": "Winter Wonderland"  },  {  "gameId": "magicquest\_ld",  "gameName": "Magic Quest"  },  {  "gameId": "sherlock",  "gameName": "Sherlock"  },  {  "gameId": "godoffortune\_ld",  "gameName": "God of Fortune"  },  {  "gameId": "klassik\_ld",  "gameName": "KlassiK"  },  {  "gameId": "lanternfestival",  "gameName": "Lantern Festival"  },  {  "gameId": "candylicious\_ld",  "gameName": "Candylicious"  },  {  "gameId": "florasecret",  "gameName": "Flora's Secret"  },  {  "gameId": "piratestreasure",  "gameName": "Pirate's Treasure"  },  {  "gameId": "legendofnezha",  "gameName": "The Legend Of Nezha"  },  {  "gameId": "panda",  "gameName": "Panda"  },  {  "gameId": "streetbasketball",  "gameName": "Street Basketball"  },  {  "gameId": "zodiac\_ld",  "gameName": "Zodiac"  },  {  "gameId": "KingsofHighway",  "gameName": "King of Highway"  },  {  "gameId": "golftour",  "gameName": "Golf Tour"  }  ]  } |

* + 1. GetAgentBankAccount ( GET )

**Mount Point:** / FrontAPI/SystemSetting.svc/accountType/{accountType}/agentBankAccount

This API is to look up a list of bank account when a player request a deposit

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accountType | mandatory | Byte | 1: deposit (default)  2: withdraw |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| bankAccountList | mandatory | list<AgentBankAccount> |  |

Agent Bank Account

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agentBankAcctSeqNo | mandatory | Int32 | agent bank account sequence number |
| bankNo | mandatory | Int32 | bank number |
| bankNm | mandatory | string | bank name written by agent admin |
| bankHolder | mandatory | string | bank holder name |
| bankAccountNo | mandatory | string | bank account number |

Example

|  |
| --- |
| {  "bankAccountList": []  } |

* 1. Operation
     1. GetAnnouncement (POST)

**Mount Point:** / FrontAPI/Operation.svc/announcements

This API is to retrieve a list of announcement.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| announceTypeCd | mandatory | Byte | 1: notifications  2: event  3: FAQ |
| isPopup | mandatory | bool | true: Popup announcement, otherwise false |
| incluedParent | optional | bool | whether parent agent’s announcement is to be displayed or not  default: **true**, otherwise: **false** |
| languageNo | optional | int | language number  default: english |

Example

|  |
| --- |
| {  "announceTypeCd":1,  "incluedParent":true,  "isPopup":true,  "languageNo":100  } |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| announcementList | mandatory | list<Announcement> |  |

Announcement

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| announceNo | mandatory | int | announcement number |
| announceTypeCd | mandatory | Byte | 1: notifications  2: event  3: FAQ |
| announceType | mandatory | string | announcement type |
| title | mandatory | string | announcement title |
| isPopup | mandatory | bool | whether parent agent’s announcement is to be displayed or not  default: true, otherwise: false |
| updateDate | mandatory | string | date format  ISO\_8601: 1981-02-22T09:00:00 |

Example

|  |
| --- |
| {  "annuncementList": [  {  "announceNo": 2,  "announceTypeCd": 1,  "announceType": "Notice",  "title": "TEST 2",  "isPopup": true,  "updateDate": "2015-06-05T08:04:02”  },  {  "announceNo": 4,  "announceTypeCd": 1,  "announceType": "Notice",  "title": "TEST 2",  "isPopup": true,  "updateDate": "2015-06-05T11:16:24”  }  ]  } |

* + 1. GetQuestionAnswerList (POST)

**Mount Point:** /FrontAPI/Operation.svc/qstAnsList

This API is to retrieve a list of Question and Answer List (OneToOne inquiry, Partner contact).

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |
| qstAnsTypeCd | mandatory | byte | question type code  1: Player 1 to 1  2: Partner |
| fromDate | mandatory | string | query start date  ISO\_8601: 1981-02-22T09:00:00 |
| toDate | mandatory | string | query end date  ISO\_8601: 1981-02-22T09:00:00 |
| languageNo | optional | int | language number |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| QAList | mandatory | list<QA> | Question Answer list |

QA

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| QstAnsNo | mandatory | Int32 | question answer number |
| parentQstAnsNo | mandatory | Int32 | parent question answer number |
| qstAnsTypeCd | mandatory | Byte | question answer type code  1: Player 1 to 1  2: Partner |
| qstAnsType | mandatory | string | question answer type |
| title | mandatory | string | announcement title |
| contents | mandatory | string | announcement contents |
| nickname | mandatory | string | player nickname left the question and answer. |
| updateDate | mandatory | string | datetime UTC |
| updateAdmId | mandatory | string | admin id or “PLAYER” |
| qstAnsLevel | optional | Int32 | question answer order number in same thread  If a player ask a question, qstAnsLevel is 0.  Then admin leave a comment on that question, qstAnsLevel is 1.  If a plyayer ask a question about above comment or an admin leave a additional comment on that question, qstAnsLevel is 2. |

Example

|  |
| --- |
| {  "questionAnswerList": [  {  "qstAnsNo": 6,  "parentQstAnsNo": 0,  "qstAnsTypeCd": 1,  "qstAnsType": "Player 1 to 1",  "title": "TEST 2 Edit",  "contents": "TEST MEMO 22222 Edit",  "nickname": "Test3",  "updateDate": “2015-06-06T02:28:32”  "updateAdmId": "PLAYER",  "qstAnsLevel": 0  },  {  "qstAnsNo": 7,  "parentQstAnsNo": 6,  "qstAnsTypeCd": 1,  "qstAnsType": "Player 1 to 1",  "title": "TEST 2",  "contents": "TEST MEMO",  "nickname": "Test1",  "updateDate": “2015-06-06T04:22:52”  "updateAdmId": "AL\_10000",  "qstAnsLevel": 1  },  ]  } |

* + 1. LeaveQuestion (POST)

**Mount Point:** / FrontAPI/Operation.svc/leaveQuestion

This API is to leave a question by agent player from Front Web.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| nickname | mandatory | string | nickname |
| parentQstAnsNo | optional | Int32 | parent question answer number |
| qstAnsTypeCd | mandatory | Byte | 1: Player 1 to 1  2: Partner |
| title | mandatory | string | question title |
| contents | mandatory | string | question contents |

Example

|  |
| --- |
| {  “nickname":"syoon4",  "qstAnsNo":100,  "qstAnsTypeCd":1,  "title":"syoon title",  "contents":"syoon contents"  } |

***Response***

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | Bad Request |  |

* + 1. GetAnnounceContents (GET)

**Mount Point:** /FrontAPI/Operation.svc/announceNo/{**announceNo**}/contents

This API is to get the contents of the announcement.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| announceNo | optional | string | announcement number |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| contents | mandatory | string | announcement contents |

Example

|  |
| --- |
| {  "contents": "Contents Test"  } |

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | Bad Request |  |

* + 1. **SendFrontWebSms** (POST)

**Mount Point:** / FrontAPI/Operation.svc/sendSms

This API is to send a SMS to authenticate the player from Front Web.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| countryCallingCd | mandatory | Int16 | country calling cd |
| tel | mandatory | string | telephone number |
| languageIsoCd | mandatory | string | language ISO code |
| message | mandatory | string | message |

Example

|  |
| --- |
| {  "countryCallingCd":82,  "tel":"01223465789",  "languageIsoCd":"ENG",  "message":"message"  } |

***Response***

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | Bad Request |  |

* 1. Marketing
     1. GetPlayerCouponHistory (GET)

**Mount Point:** / FrontAPI/Marketing.svc/token/{**token**}/couponHistory

This API is to look up a used or not used coupon list of agent.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| couponList | mandatory | list<Coupon> |  |

Coupon

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| couponCode | mandatory | string | coupon code |
| currencyAmount | mandatory | CurrencyAmount | coupon currency amount |
| couponName | mandatory | string | coupon name |
| expirationDate | optional | string | UTC time  ISO\_8601: 1981-02-22T09:00:00 |
| status | mandatory | string | coupon status |

CurrencyAmount

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyIsoCd | optional | string | currency ISO code |
| amount | mandatory | decimal | coupon amount |

Example

|  |
| --- |
| {  "CouponList": [  {  "CouponCode": "28DDYKM585V7",  "CouponName": "Test Coupon Bonus 3 @",  "CurrencyAmount": {  "currencyIsoCd": "USD",  "amount": 200  },  "ExpirationDate": "",  "Status": "Issued"  },  {  "CouponCode": "W97FT2982GD4",  "CouponName": "Test Coupon Bonus 3 @",  "CurrencyAmount": {  "currencyIsoCd": "USD",  "amount": 200  },  "ExpirationDate": "",  "Status": "Redeemed"  },  ]  } |

* + 1. UseCoupon (PUT)

**Mount Point:** /FrontAPI/Marketing.svc/useCoupon/{**accessToken**}/{**couponCode**}

This API is to use a coupon.

If available coupon, player’ balance will be updated.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |
| couponCode | mandatory | string | coupon code |

* 1. Agent
     1. GetAgentContactInfo (GET)

**Mount Point:** /FrontAPI/Agent.svc/contactInfo

This API is to retrieve an agent’s contact information.

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| contactEmailList | mandatory | List<string> | contact email list |
| contactPhoneList | mandatory | List<string> | contact phone list |
| contactSnsList | mandatory | List<SnsInfo> | contact SNS list |

SnsInfo

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| snsName | mandatory | string | sns name |
| snsId | mandatory | string | sns id |

Example

|  |
| --- |
| {  "contactEmailList": [  "test1@test1.com1",  "test2@test2.com2"  ],  "contactPhoneList": [  "123-456-789",  "098-765-432"  ],  "contactSnsList": [  {  "snsName": "Skype",  "snsId": "testMsgr1"  },  {  "snsName": "Nate",  "snsId": "test2@test2.com2"  }  ]  } |

* 1. FraudAndRisk
     1. GetAgentBlackIpList (GET)

**Mount Point:** / FrontAPI/Marketing.svc/agentBlackIpList

This API is to look up a black IP list regarding to an agent.

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| ipBlacklist | mandatory | list<string> | IP black list |

Example

|  |
| --- |
| {  "IpBlackList": [  "10.10.10.111",  "10.10.10.123"  ]  } |

# **Main Wallet API**

**Content-Type: application/json**

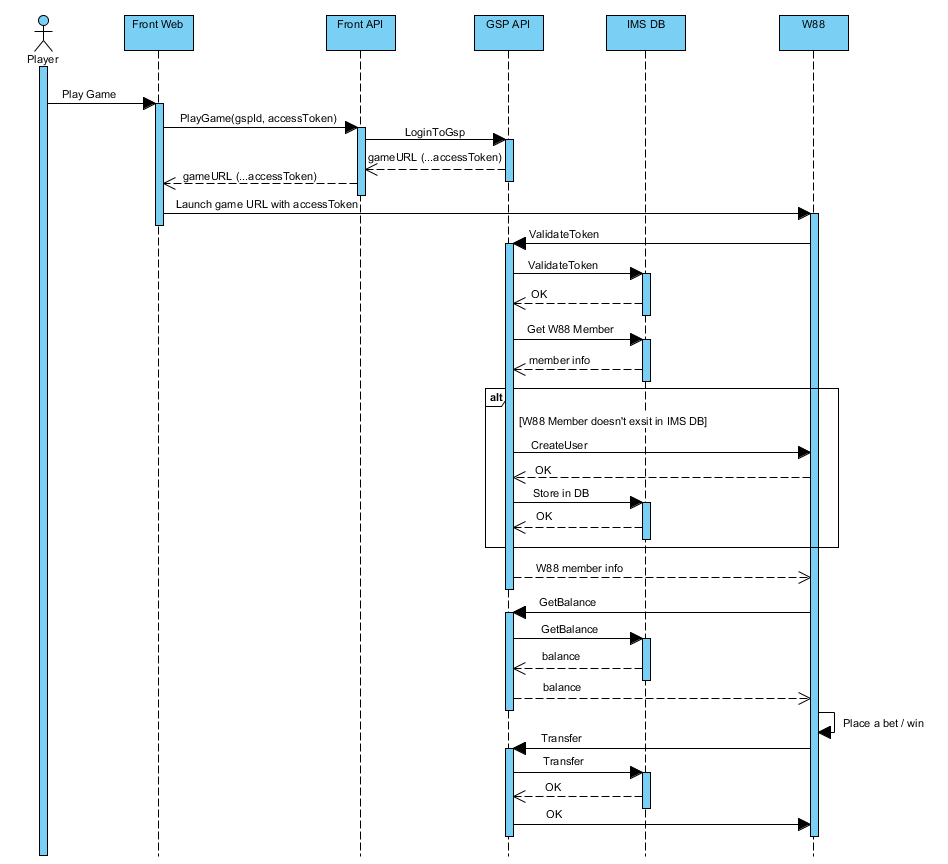
Request Header information

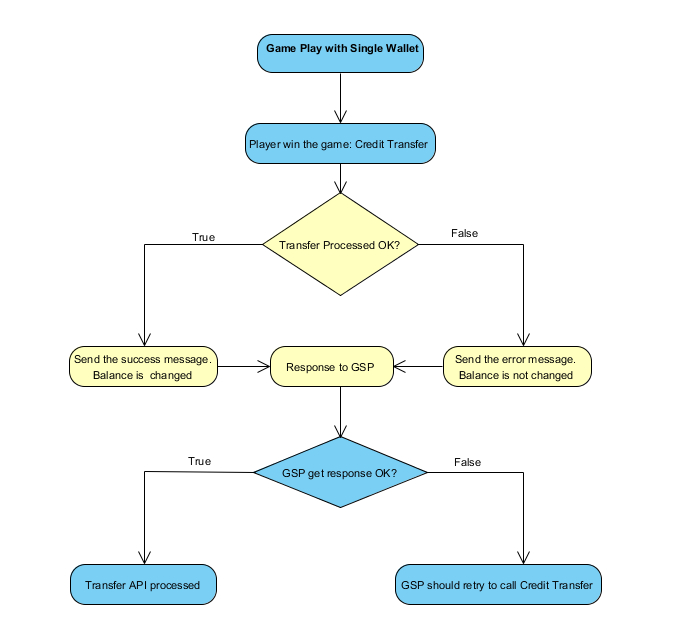
|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **Header** |
| opId | mandatory | string | Gsp id |
| opPassword | mandatory | string | Gsp password |
| method | mandatory | string | method name |

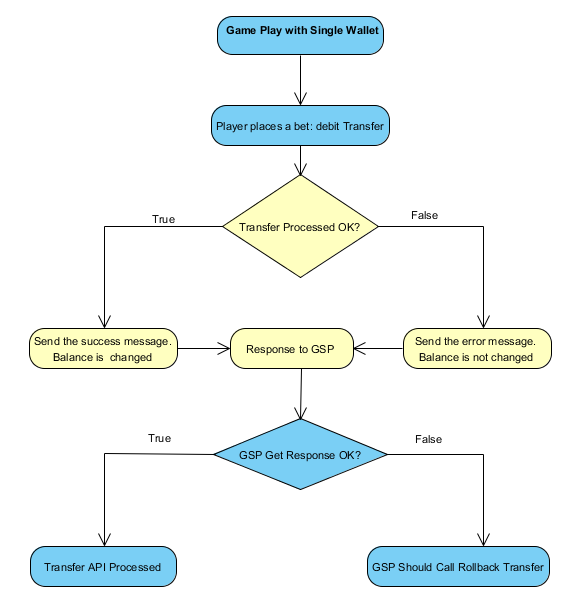
Response Header information

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **Header** |
| responseCode | mandatory | int | response code |
| errorMessage | mandatory | string | error message |

* 1. GamePlay (GamePlay)







content-type: application/xml

|  |  |  |  |
| --- | --- | --- | --- |
| **Reference Document** | **Version** | **File Name** | **API** |
| Authentication & Wallet APIs | 1.02 | GamePlayIntAuthenticationWalletAPIsv1.02.pdf | ValidateToken  CreateUser  PlayGame – LiveCasino  PlayGame – 3DSlots |
| GAMEPLAY Slots Offering | 1.02 | GamePlayIntSlotsofferingv1.02.docx.pdf | CTXM, BETSOFT, AGS, PNG |

* + 1. ValidateToken ( GET ) – From W88 to IMS

**Mount Point:** /GspService/W88.svc/token/{**accessToken**}/validate

This API is to validate the token which is included in launched game URL.



#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| token | mandatory | varchar(128) | token via game url |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error\_code | mandatory | int(11) | error code |
| error\_msg | optional | varchar(256) | error description |
| cust\_id | mandatory | varchar(100) | gspMember id |
| cust\_name | mandatory | varchar(100) | gspMember id  ( Alphanumeric only, do not use Chinese, Korean, Thai or Vietnam characters) |
| currency\_code | mandatory | varchar(16) | gspMember currency ISO code |
| language | mandatory | varchar(16) | gspMember Browser language code  default: ??  Currency Code Description Language  RMB Chinese Yuan zh-cn  IDR Indonesia Rupiah id-id  MYR Malaysia Ringgit ??  THB Thai Baht th-th  USD US Dollar km-kh  VND Vietnam Dong vi-vn  KRW Korean Won ko-kr |
| country | mandatory | varchar(16) | gspMember Country |
| ip | mandatory | varchar(16) | gspMember login ip address |
| date\_of\_birth | mandatory | varchar(16) | gspMember birth date(dd-mm-yyyy) |
| test\_cust | mandatory | bool | is test user  default: false |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <resp>  <error\_code>0</error\_code>  <cust\_id>10001</cust\_id>  <cust\_name>Dummy</cust\_name>  <currency\_code>IDR</currency\_code>  <language>en-us</language>  <test\_cust>false</test\_cust>  <country>USA</country>  <date\_of\_birth>29-09-1989</date\_of\_birth>  <ip>1.1.1.1</ip>  </resp> |
|  |

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <resp>  <error\_code>03</error\_code>  <error\_msg> Authentication Failed </error\_msg>  </resp> |

* + 1. CreateMember (POST)

**Mount Point: ~** **/op/createuser**

The API is used to create a member.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| merch\_id | mandatory | varchar(100) | Gsp id |
| merch\_pwd | mandatory | varchar(100) | Gsp password |
| cust\_id | mandatory | varchar(100) | gspMember id |
| cust\_name | mandatory | varchar(100) | gspMember id |
| currency | mandatory | varchar(16) | gspMember’s currency ISO code  RMB Chinese Yuan  IDR Indonesia Rupiah  MYR Malaysia Ringgit  THB Thai Baht  USD US Dollar  VND Vietnam Dong  KRW Korean Won |
| test\_cust | optional | Boolean | true: test member account  false: default, real member account |

Example

|  |
| --- |
| http://club8api.bet8uat.com/op/createuser?merch\_id=&merch\_pwd=&cust\_id=test01&cust\_name=test01&currency=  MYR |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error\_code | mandatory | int(11) | error code |
| error\_msg | optional | varchar(256) | error description |

Error Code Detail

|  |
| --- |
| 0 No errors  1 Transaction already exists  -1 Unknown error  -2 Player not found  -3 Ticket not found or expired  -4 Insufficient balance  -7 Player is blocked  -27 Parameter type mismatch  -29 Currency mismatch  -33 Invalid amount  -48 Invalid merchant password  -119 Missing input parameter  -203 Customer already exist |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
|  |

* + 1. GetBalance ( GET )

**Mount Point:** /GspService/Finance.svc/token/{**accessToken**}/balance

This API is to look up player’s balance in Single Wallet.



#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| balance | mandatory | List<Balance> | Does it support multi-currency? |

Balance

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| currencyIsoCd | optional | string | currency ISO code |
| amount | mandatory | numeric(19,2) | the latest player’s balance amount |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |

Example

|  |
| --- |
| {  "currencyIsoCd": "USD",  "balance": 1108.95  } |

* + 1. PlayGame

Mount Point: it depends on game type?

The API is used to play the game.

* + 1. LiveCasino

Web: http://casino.w88uat.com?token=**accessToken**&m=**modeOfView**&op=**gspId**&lang=**languageIsoCd** &tb=**table**

Mini: http://casino.w88uat.com/minitable?token=**accessToken**&op=**gspId**&lang=**languageIsoCd**&tb=**table**

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |
| modeOfView | optional | string | mode of view: 2d, 3d, m, 7s  default: 2d |
| gspId | mandatory | string | gsp id  default: W88 |
| languageIsoCd | optional | string | Web Browser Language Identification Codes (LIC)  default: en-us  en-us, id-id, ja-jp, km-kh,ko-kr, th-th, vi-vn, zh-cn |
| table | optional | string | only applicable when modeOfView = 2d or 3d  default: 0  1 Baccarat  2 Baccarat  3 Baccarat  4 Dragon Tiger  5 Sicbo  6 Roulette  7 7UP Baccarat  8 3 Pictures  9 Super Color Sicbo  10 Blackjack  11 Super Ball/Tambola  12 Super Fan Tan  101 Super98/Non Commission Baccarat  102 Super98/Non Commission Baccarat  103 Super98/Non Commission Baccarat |

Example

|  |
| --- |
|  |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameUrl | mandatory | string | game url |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
|  |

* + - 1. 3DSlots

Web: http://slots.w88uat.com?token=**accessToken**&gameid=**gameId**&fun=**fun**&op=**gspId**&lang= **languageIsoCd**

Mobile: http://mslots.globalintgames.com/**gameId**/?fun=**fun**&token=**accessToken**&op=**gspId**

Mini:

http://slots.w88uat.com/menu\_mini.swf?op=**gspId**&gameid=**gameId**&fun=**fun**&info=**info**

&token=**accessToken**&lang=**languageIsoCd**

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |
| gameId | mandatory | string | game to load |
| fun | mandatory | int | 0: real play (default)  1: fun play |
| gspId | mandatory | string | gsp id  default: W88 |
| languageIsoCd | optional | string | Web Browser Language Identification Codes (LIC)  default: en-us  en-us, id-id, ja-jp, km-kh,ko-kr, th-th, vi-vn, zh-cn |
| info | optional | string | rate: show chip price  credit: show balance (default) |
| switch | optional | int | 0: allow switching (default)  1: disable switching |
| hidelobby | optional | int | 0: show (default)  1: hide |

Example

|  |
| --- |
|  |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameUrl | mandatory | string | game url |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
|  |

* + - 1. BetSoft

Web & Mobile:

http://casino.w88uat.com/mini/betSoft.html?op=**gspId**&token=**accessToken**&mode=**playMode**&lang= **languageIsoCd**&gameId=**gameCode**

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gspId | mandatory | string | gsp id |
| accessToken | mandatory | string | access token |
| playMode | mandatory | int | 0: real play (default)  1: fun play |
| languageIsoCd | manatory | string | language ISO code alpha-2 |
| gameCode | optional | string | game code |

Example

|  |
| --- |
|  |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameUrl | mandatory | string | game url |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
|  |

* + - 1. Top Trend Gaming (TTG)
      2. CTXM
      3. Play N Go (PNG)
    1. PostTransfer( POST )

**Mount Point:** / FrontAPI/Finance.svc/postTransfer

This API is to process player’s money transaction.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gspId | optional | Long | game service provider id  it will be handled by API. |
| transactionNo | mandatory | Long | transaction number |
| accessToken | mandatory | string | access token  Used if TransactionCode is 11005 (Debit) |
| gspMemberId | mandatory | string | gspMember id  Used if TransactionCode is 11006 (Credit) |
| cashAmount | mandatory | numeric(19,2) | cash amount  Debit: only negative (-) is accepted  Credit: only positive (+) is accepted  Not called with zero (0) amount |
| transactionSourceCode | mandatory | Int | transaction source code  Reference in “TransactonCode” document |
| transactionTypeCode | mandatory | Int | transaction type code  Reference in “TransactonCode” document |
| transactionDescription | mandatory | Int | transaction description  Reference in “TransactonCode” document |
| effectiveDate | mandatory | dateTime | The time that the API is called.  Unix epoch time in milliseconds |
| GameType | mandatory | string | game type  Reference in “GameCode” document |
| transactionClassCode | mandatory | Int | 101: Poker  102: Casino |

Example

|  |
| --- |
| {  "transactionNo":2,  "accessToken":"EC49F0AC835846F2ABADD2481533333A",  "externalId":"",  "cashAmount":123.45,  "transactionSourceCode":1101,  "transactionTypeCode":11006,  "transactionDescription":120002,  "effectiveTime":"\/Date(928117200000+0900)\/",  "gameType":"Omaha",  "gameEntryNo":922337203,  "cashGameInstanceNo":922337203,  "tournamentInstanceNo":922337203,  "transactionClassCode":101  } |

#### Response

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |

* + 1. RollbackTransfer ( POST )

**Mount Point:** / FrontAPI/Finance.svc/rollbackTransfer

This API is to rollback a transaction.



#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agentId | mandatory | string | agent id |
| playerId | mandatory | string | playerId |
| transactionId | mandatory | string | transaction id |
| startDate | mandatory | string | start date |
| endDate | mandatory | string | end date |

Example

|  |
| --- |
|  |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| totalCredits | mandatory | numeric(19,2) | Sum of credit amount during the searched term |
| totalDebits | mandatory | numeric(19,2) | Sum of debit amount during the searched term |
| netActivity | mandatory | numeric(19,2) | totalCredits – totalDebits |
| endingBalance | mandatory | numeric(19,2) | Current cash balance |
| cashTransactionDetail | mandatory | List | List of player’s cash transaction history |

Cash Transaction Detail list

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| transactioDate | mandatory | Datetime | transaction date |
| transactionDescription | mandatory | string | transaction description |
| cashAmount | mandatory | numeric(19,2) | transaction cash amount |
| balanceAfter | mandatory | numeric(19,2) | cash amount after this transaction |
| balanceBefore | mandatory | numeric(19,2) | Balance prior to applying the transaction |
| transactionClass | mandatory | string | transaction class: Poker, Casino |
| transactionSource | mandatory | string | transaction source  Reference in “TransactonCode” document |
| transactionType | mandatory | string | transaction type |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **RESPONSE CODE** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| {  "totalCredits": 740.7,  "totalDebits": -2.1,  "netActivity": 742.8,  "endingBalance": 1232.4,  "cashTransactionDetail": [  {  "transactioDate": "\/Date(1419424958153+0000)\/",  "transactionDescription": "Buy-In",  "cashAmount": 123.45,  "balanceAfter": 493.8,  "balanceBefore": 617.25,  "transactionClass": "Poker",  "transactionSource": "Cash Game Server",  "transactionType": "Credit"  },  {  "transactioDate": "\/Date(1419425109173+0000)\/",  "transactionDescription": "Buy-In",  "cashAmount": 123.45,  "balanceAfter": 617.25,  "balanceBefore": 740.7,  "transactionClass": "Poker",  "transactionSource": "Cash Game Server",  "transactionType": "Credit"  },  {  "transactioDate": "\/Date(1419425116043+0000)\/",  "transactionDescription": "Buy-In",  "cashAmount": 123.45,  "balanceAfter": 740.7,  "balanceBefore": 864.15,  "transactionClass": "Poker",  "transactionSource": "Cash Game Server",  "transactionType": "Credit"  },  {  "transactioDate": "\/Date(1419553396927+0000)\/",  "transactionDescription": "Buy-In",  "cashAmount": -0.7,  "balanceAfter": 987.6,  "balanceBefore": 986.9,  "transactionClass": "Poker",  "transactionSource": "Cash Game Server",  "transactionType": "Debit"  },  ]  } |

* + 1. CheckTransactionResult (POST)

**Mount Point: ~** **/op/check**

The API is used to check whether the transaction is executed successfully or not.

Q: Is there a reason to call the API before using Credit() not after invoking Credit() ?

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| merch\_id | mandatory | varchar(100) | Gsp id |
| merch\_pwd | mandatory | varchar(100) | Gsp password |
| trx\_id | mandatory | varchar(128) | Unique transaction ID from operator |

Example

|  |
| --- |
|  |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error\_code | mandatory | int(11) | error code |
| error\_msg | optional | varchar(256) | error description |
| before | optional | decimal(18,2) | gspMember balance before transaction |
| after | optional | decimal(18,2) | gspMember balance after transaction |
| exchange\_rate | optional | decimal(18,2) | exchange rate to RMB? |
| trx\_id | optional | decimal(18,2) | next transaction id created from W88 |

Example

|  |
| --- |
|  |

* + 1. GetLiveCasinoBetHistory (GET)

**Mount Point:** [**https://csnbo**](https://csnbo)**.w88.com/csnbo/api/op/betDetail.html?merch\_id={merch\_id}&page\_num={page\_num}&page\_size={page\_size}&merch\_pwd={merch\_pwd}&date\_from={date\_from}&date\_to={date\_to}**

The API is used to save the detailed bet history ( which is retrieved from W88 Database ) to Hub Database.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| merch\_id | mandatory | string | Gsp id |
| page\_num | mandatory | string | page number: 1(hard-code) |
| page\_size | mandatory | string | page size: 100(hard-code)  default: 100 MAX: 1000 |
| merch\_pwd | mandatory | string | Gsp password |
| date\_from | mandatory | string | Query date from  UTC time( yyyy-MM-dd HH:mm:ss ) |
| date\_to | mandatory | string | Query date to  UTC time( yyyy-MM-dd HH:mm:ss ) |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errorCode | mandatory | string | error code |
| pageNum | mandatory | string | page number |
| pageSize | mandatory | string | page size |
| totalRow | mandatory | string | total row number |
| totalPage | mandatory | string | total page number |
| item | mandatory | List<W88LiveCasinoHistory> | bet amount |

W88 Live Casino History

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| transId | mandatory | string | transaction id |
| status | mandatory | string | transaction status  1: Normal 2: Void |
| transDate | mandatory | string | transaction date |
| userId | mandatory | string | gspMember id |
| gameType | mandatory | string | game description |
| bet | mandatory | decimal | bet amount |
| winlose | mandatory | decimal | result amount of bet |
| balance | mandatory | decimal | member’s balance after bet transaction |
| gameResult | mandatory | string | game result |
| currency | mandatory | string | member’s currency |

* + 1. GetCtxmBetHistory (GET)

**Mount Point:** [**https://club**](https://club)**8api.w88.com/ms/betDetail?merch\_id={merch\_id}&merch\_pwd={merch\_pwd} &date\_from={date\_from}&date\_to={date\_to}&page\_num={merch\_id}&game\_provider={game\_provider}**

Ctxm ??

DTO is similar to SaveLiveCasinoHistory except game\_provider.

The API is used to save the detailed bet history ( which is retrieved from W88 Database ) to Hub Database.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| merch\_id | mandatory | string | Gsp id |
| merch\_pwd | mandatory | string | Gsp password |
| date\_from | mandatory | string | Query date from  UTC time( yyyy-MM-dd HH:mm:ss ) |
| date\_to | mandatory | string | Query date to  UTC time( yyyy-MM-dd HH:mm:ss ) |
| page\_num | mandatory | string | page number: 1(hard-code) |
| game\_provider | mandatory | string | game provider: CTXM(hard-code) |

#### Response

Response Parameters

DTO is same as SaveLiveCasinoHistory.

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errorCode | mandatory | string | error code |
| pageNum | mandatory | string | page number |
| pageSize | mandatory | string | page size |
| totalRow | mandatory | string | total row number |
| totalPage | mandatory | string | total page number |
| item | mandatory | List<W88CtxmHistory> | bet amount |

W88 Ctxm History

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| transId | mandatory | string | transaction id |
| status | mandatory | string | transaction status  1: Normal 2: Void |
| transDate | mandatory | string | transaction date |
| userId | mandatory | string | gspMember id |
| gameType | mandatory | string | game description |
| bet | mandatory | decimal | bet amount |
| winlose | mandatory | decimal | result amount of bet |
| balance | mandatory | decimal | member’s balance after winlose? |
| gameResult | mandatory | string | game result |
| currency | mandatory | string | gspMember’s currency ISO code |

* + 1. **Get3DSlotBetHistory** (GET)

**Mount Point:** [**http://slotservice**](http://slotservice)**.gpiops.com/integration.asmx/gettransactionhistory?startDate={startDate}** **&endDate={endDate}&pageNum={pageNum}&pageSize={pageSize}&operator={operator}&pwd={pwd}**

The API is used to save the detailed bet history ( which is retrieved from W88 Database ) to Hub Database.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| startDate | mandatory | string | query start date |
| endDate | mandatory | string | query end date |
| pageNum | mandatory | string | page number: 1(hard-code) |
| pageSize | mandatory | string | page size: 100(hard-code)  default: 100 MAX: 1000 |
| operator | mandatory | string | Gsp id |
| pwd | mandatory | string | Gsp password2 |

#### Response

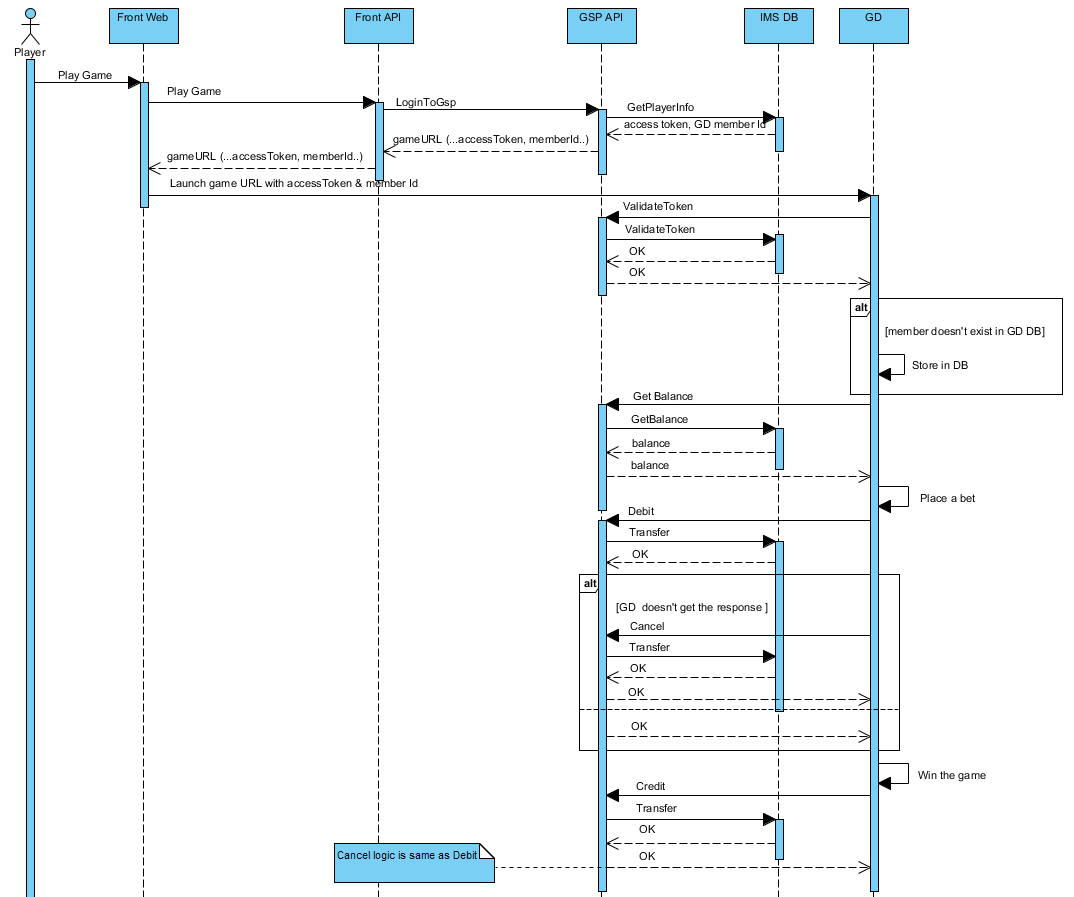
Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errorCode | mandatory | string | error code |
| pageNum | mandatory | string | page number |
| pageSize | mandatory | string | page size |
| currentRows | mandatory | string | current rows |
| totalRow | mandatory | string | total row number |
| totalPage | mandatory | string | total page number |
| history | mandatory | List<W883DSlotHistory> | bet amount |

W88 3D Slot History

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| operationCode | mandatory | string | operation code? |
| userId | mandatory | string | game service member id |
| changeTime | mandatory | string | change time? |
| changeType | mandatory | string | change type? |
| gameType | mandatory | string | game description |
| gameName | mandatory | string | game name |
| bet | mandatory | decimal | bet amount |
| ret | mandatory | decimal | ret? |
| changes | mandatory | decimal | winloss amount |
| endBalance | mandatory | decimal | balance after bet transaction |
| trxId | mandatory | string | bet transaction id |
| operator | mandatory | string | Gsp id |
| ver | mandatory | string | ver |

* 1. Gold Deluxe (GD) – 컨펌 대기중



|  |  |  |  |
| --- | --- | --- | --- |
| **Reference Document** | **Version** | **File Name** | **API** |
| GD Integration: Single Wallet Specification | 1.3 | GD single wallet API\_v1\_3.doc | ValidateToken  CreateUser  PlayGame – LiveCasino  PlayGame – 3DSlots |
| GAMEPLAY Slots Offering | 1.02 | GamePlayIntSlotsofferingv1.02.docx.pdf | CTXM, BETSOFT, AGS, PNG |

Currently Merchant’s system supports only HTTP SOAP protocol.

The network communication will be in XML format and use HTTP(S) POST request.

The SOAP server function name must be same as the following name

(GetUserBalance, Debit, Cancel, Credit)

* + 1. CreateMember

gspMember account is created when player login into GD Casino.

* + 1. ValidateToken ( GET ) - TBD
    2. GetMemberBalance

**Mount Point: ~/**GetUserBalance

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| userID | mandatory | string | gspMember id |
| currency | mandatory | string | gspMember currency ISO code |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://testnamespace.org">  <SOAP-ENV:Body>  <ns1:GetUserBalance>  <ns1:userId>tonysin</ns1:userId>  <ns1:currency>CNY</ns1:currency>  </ns1:GetUserBalance>  </SOAP-ENV:Body>  </SOAP-ENV:Envelope> |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| StatusCode | mandatory | int | indicates success / failure |
| UserBalance | mandatory | double | player balance after transaction |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://testnamespace.org" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">  <SOAP-ENV:Body>  <ns1:GetUserBalanceResponse>  <StatusCode xsi:type="xsd:int">0</StatusCode>  <UserBalance xsi:type="xsd:float">88888888</UserBalance>  </ns1:GetUserBalanceResponse>  </SOAP-ENV:Body>  </SOAP-ENV:Envelope> |

* + 1. Credit

**Mount Point:** ~/Credit

The API is used to credit transfer when player win the game.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| userId | mandatory | string | gspMember Id |
| gameId | mandatory | string | Game id for which the withdraw is requested,  e.g. B713050301303  Game ID for parlay bet will show “Parlay” |
| gameType | mandatory | string | Game type e.g.:  6 - Baccarat;  26- Blackjack  28 - Roulette;  30 - Muti-Bet;  100 – Slot game |
| transactionId | mandatory | string | Unique transaction id that identifies the request uniquely. This transaction id is same when the request is resent from GD system. From Merchant’s system there shouldn’t be multiple credit from player account if this same request is tried with same transaction id.  32bits-length string is generated by server randomly, e.g. 130503153010aYlmzpg3VqhB8hrMt0Xo |
| amount | mandatory | double | Amount that need to credit to player account. |
| currency | mandatory | string | Currency code eg: CNY, KRW |
| validBetAmount | mandatory | double | valid bet amount which is not refund or cancel in this transaction |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?><SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/  envelope/" xmlns:ns1="http://testnamespace.org">  <SOAP-ENV:Body>  <ns1:Credit>  <ns1:userId>tims3\_m1</ns1:userId>  <ns1:gameId>B11412033326</ns1:gameId>  <ns1:gameType>6</ns1:gameType>  <ns1:transactionId>141203153150Nk4AVHoHwEjqJoPUIcjB</ns1:transactionId>  <ns1:amount>000.00</ns1:amount>  <ns1:currency>KRW</ns1:currency>  <ns1:validBetAmount>020.00</ns1:validBetAmount>  </ns1:Credit>  </SOAP-ENV:Body>  </SOAP-ENV:Envelope> |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| StatusCode | mandatory | int | indicates success / failure |
| UserBalance | mandatory | double | player balance after transaction |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://testnamespace.org" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">  <SOAP-ENV:Body>  <ns1:CreditResponse>  <StatusCode xsi:type="xsd:int">0</StatusCode>  <UserBalance xsi:type="xsd:string">28000.0000</UserBalance>  </ns1:CreditResponse>  </SOAP-ENV:Body>  </SOAP-ENV:Envelope> |

* + 1. Debit

**Mount Point:** ~/Debit

The API is used to debit transfer when player places a bet.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| userId | mandatory | string | gspMember Id |
| gameId | mandatory | string | game id for which the debit is requested,  e.g. B713050301303  Game ID for parlay bet will show “Parlay” |
| gameType | mandatory | string | Game type e.g.:  6 - Baccarat;  26- Blackjack  28 - Roulette;  30 - Muti-Bet;  100 – Slot game |
| transactionId | mandatory | int | Unique transaction id that identifies the request uniquely. This transaction id is same when the request is resent from GD system. From Merchant’s system there shouldn’t be multiple debit from player account if this same request is tried with same transaction id.  32bits-length string is generated by server randomly, e.g. 130503153010aYlmzpg3VqhB8hrMt0Xo |
| amount | mandatory | double | Amount that need to debit from player account. |
| currency | mandatory | string | currency ISO code |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?><SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/  envelope/" xmlns:ns1="http://testnamespace.org">  <SOAP-ENV:Body>  <ns1:Debit>  <ns1:userId>tims3\_m1</ns1:userId>  <ns1:gameId>B21412034606</ns1:gameId>  <ns1:gameType>6</ns1:gameType>  <ns1:transactionId>141203182550KtkwPBoyUiouDt4zAYJR</ns1:transactionId>  <ns1:amount>2000.00</ns1:amount>  <ns1:currency>KRW</ns1:currency>  </ns1:Debit>  </SOAP-ENV:Body>  </SOAP-ENV:Envelope> |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| StatusCode | mandatory | int | indicates success / failure |
| UserBalance | mandatory | double | player balance after transaction |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://testnamespace.org" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">  <SOAP-ENV:Body>  <ns1:DebitResponse>  <StatusCode xsi:type="xsd:int">0</StatusCode>  <UserBalance xsi:type="xsd:string">26000.0000</UserBalance>  </ns1:DebitResponse>  </SOAP-ENV:Body>  </SOAP-ENV:Envelope> |

* + 1. Cancel

**Mount Point:** ~/Cancel

The API is used to cancel a transaction for refund.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| userId | mandatory | string | gspMember Id |
| gameId | mandatory | string | game id for which the debit is requested,  e.g. B713050301303  Game ID for parlay bet will show “Parlay” |
| gameType | mandatory | string | Game type e.g.:  6 - Baccarat;  26- Blackjack  28 - Roulette;  30 - Muti-Bet;  100 – Slot game |
| transactionId | mandatory | int | Unique transaction id that identifies the request uniquely. This transaction id is same when the request is resent from GD system. From Merchant’s system there shouldn’t be multiple debit from player account if this same request is tried with same transaction id.  32bits-length string is generated by server randomly, e.g. 130503153010aYlmzpg3VqhB8hrMt0Xo |
| amount | mandatory | double | Amount that need to debit from player account. |
| currency | mandatory | string | currency ISO code |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"  xmlns:ns1="http://testnamespace.org">  <SOAP-ENV:Body>  <ns1:Cancel>  <ns1:userId>tims3\_m1</ns1:userId>  <ns1:gameId>B21412034606</ns1:gameId>  <ns1:gameType>6</ns1:gameType>  <ns1:transactionId>141203182550KtkwPBoyUiouDt4zAYJR</ns1:transactionId>  <ns1:amount>2000.00</ns1:amount>  <ns1:currency>KRW</ns1:currency>  </ns1:Cancel>  </SOAP-ENV:Body>  </SOAP-ENV:Envelope> |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| StatusCode | mandatory | int | indicates success / failure |
| UserBalance | mandatory | double | player balance after transaction |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://testnamespace.org" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">  <SOAP-ENV:Body>  <ns1:CancelResponse>  <StatusCode xsi:type="xsd:int">0</StatusCode>  <UserBalance xsi:type="xsd:string">28000.0000</UserBalance>  </ns1:CancelResponse>  </SOAP-ENV:Body>  </SOAP-ENV:Envelope> |

* + 1. GetJackpotPool

**Mount Point:** [**http://wsciti**](http://wsciti)**.gdsecure88.com/MerchantAPI/eWallet.php**

The API is used to get all jackpot pool information with amount and increment of slot game.

#### Request

Request Parameters

Method value in Header: cGetJackpotPool

MessageID format: “J”+YYMMDDhhmmss + 5 random char

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| CurrencyCode | mandatory | string | currency ISO code |
| GameIDList | mandatory | int | RNG game ID of the slot game |

Example

|  |
| --- |
|  |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Game | mandatory | Game |  |
| ErrorDesc | mandatory | string | error message for non-zero reply |

Game

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Pool | mandatory | List<Pool> |  |
| GameID | mandatory | int | the RNG game ID of the slot game |

Pool

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Jackpot | mandatory | decimal | jackpot pool’s amount |
| Increment | mandatory | decimal | jackpot pool increment |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
|  |

* + 1. PlayGame

**Mount Point: ~?OperatorCode={operatorCode}&LoginTokenID={loginTokenID}&lang={language}&playerid={gspMemberId}**

The API is used to get a game url.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| operatorCode | mandatory | string | Gsp id |
| loginTokenID | mandatory | string | gspMember token |
| lang | mandatory | string | language ISO code  Web Browser Language Identification Codes |

Example

|  |
| --- |
|  |

#### Response

Response Parameters

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

* + 1. GetBetHistory

**Mount Point:** [**http://wsciti**](http://wsciti)**.gdsecure88.com/MerchantAPI/eWallet.php**

The API is used to get the list of Game Bet Transaction History from GD Database.

#### Request

Request Parameters

Get Bet History response message format: (number of record return limited 500).

Please change the <index> tag in request API to “501” to get the bet record after 500 in the same time period.

Method value in Header: cGetBetHistory

MessageID format: “H”+YYMMDDhhmmss + 5 random char

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| FromTime | mandatory | string | query start time  date format: MM/DD/YYYY HH:MM:SS |
| ToTime | mandatory | string | query end time  date format: MM/DD/YYYY HH:MM:SS |
| Index | mandatory | int | start index of the records within the time period |
| UserID | mandatory | string | the specified member id for history searching.  Can be null to get all bet history for all players |
| ShowBalance | optional | string | 1: show  the other: don’t show? |

Example

|  |
| --- |
|  |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| TotalRecord | mandatory | int | total record number of this response |
| ErrorDesc | mandatory | string | error message for non-zero reply |
| BetInfo | mandatory | BetInfo | each bet information, no this part when error was found |

Bet Info

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| No | mandatory | int | record number within this time period |
| UserID | mandatory | string | gspMember id |
| BetTime | mandatory | string | spin time for processing game  complete time for completed game.  Please use the complete time for reporting. Processing bet will not show in P&L and win loss report |
| BalanceTime | mandatory | string | balance time? |
| ProductID | mandatory | string | Product ID? |
| GameInterface | mandatory | string | Game Interface |
| BetID | mandatory | string | unique ID for the bet  The BetID contains within 64 characters.  Should be able to handle 64 characters for the betID of RNG game |
| BetType | mandatory | string | bet type? |
| BetAmount | mandatory | decimal | bet amount of this bet |
| WinLoss | mandatory | decimal | total win loss including bet amount |
| BetResult | mandatory | string | bet result |
| StartBalance | mandatory | decimal | balance before bet |
| EndBalance | mandatory | decimal | balance after bet |
| BetArrays | mandatory | BetArray |  |
| time | mandatory | string | time? |
| RNGID | mandatory | string | RNG game ID |
| JackpotWin | mandatory | decimal | jackpot win amount |
| status | mandatory | string | Processing: when the game still in progress  Completed: when the game is completed |

Bet Array

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Bet | mandatory | List<Bet> |  |

Bet

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| GameID | mandatory | string | game id |
| SubBetType | mandatory | string | sub bet type |
| GameResult | mandatory | string | game result  What is the difference between game result and bet result? |
| WinningBet | mandatory | string | bet amount? |
| TableID | mandatory | string | table id |
| DealerID | mandatory | string | dealer id |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
|  |

* + 1. GetRNGBetHistory

**Mount Point:**

The API is used to get the list of RNG Game Bet Transaction History from GD Database.

#### Request

Request Parameters

Get RNG Bet History response message format: (number of record return limited 500).

Please change the <index> tag in request API to “501” to get the bet record after 500 in the same time period.

Method value in Header: cGetRNGBetHistory

MessageID format: “R”+YYMMDDhhmmss + 5 random char

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| FromTime | mandatory | string | query start time  date format: MM/DD/YYYY HH:MM:SS |
| ToTime | mandatory | string | query end time  date format: MM/DD/YYYY HH:MM:SS |
| Index | mandatory | int | start index of the records within the time period |
| UserID | mandatory | string | the specified member id for history searching.  Can be null to get all bet history for all players |
| ShowBalance | optional | string | 1: show  the other: don’t show? |

Example

|  |
| --- |
|  |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| TotalRecord | mandatory | int | total record number of this response |
| ErrorDesc | mandatory | string | error message for non-zero reply |
| BetInfo | mandatory | BetInfo | each bet information, no this part when error was found |

Bet Info

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| No | mandatory | int | record number within this time period |
| UserID | mandatory | string | gspMember id |
| BetTime | mandatory | string | spin time for processing game  complete time for completed game.  Please use the complete time for reporting. Processing bet will not show in P&L and win loss report |
| BalanceTime | mandatory | string | balance time? |
| ProductID | mandatory | string | Product ID? |
| GameInterface | mandatory | string | Game Interface |
| BetID | mandatory | string | unique ID for the bet  The BetID contains within 64 characters.  Should be able to handle 64 characters for the betID of RNG game |
| BetType | mandatory | string | bet type? |
| BetAmount | mandatory | decimal | bet amount of this bet |
| WinLoss | mandatory | decimal | total win loss including bet amount |
| BetResult | mandatory | string | bet result |
| StartBalance | mandatory | decimal | balance before bet |
| EndBalance | mandatory | decimal | balance after bet |
| BetArrays | mandatory | BetArray |  |
| time | mandatory | string | time? |
| RNGID | mandatory | string | RNG game ID |
| JackpotWin | mandatory | decimal | jackpot win amount |
| status | mandatory | string | Processing: when the game still in progress  Completed: when the game is completed |

Bet Array

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Bet | mandatory | List<Bet> |  |

Bet

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| BetID | mandatory | string | bet id  What is the difference between BetID in BetInfo and BetID in Bet? |
| GameID | mandatory | string | game id |
| SubBetType | mandatory | string | sub bet type |
| GameResult | mandatory | string | game result  What is the difference between game result and bet result? |
| WinningBet | mandatory | string | bet amount? |
| TableID | mandatory | string | table id |
| DealerID | mandatory | string | dealer id |

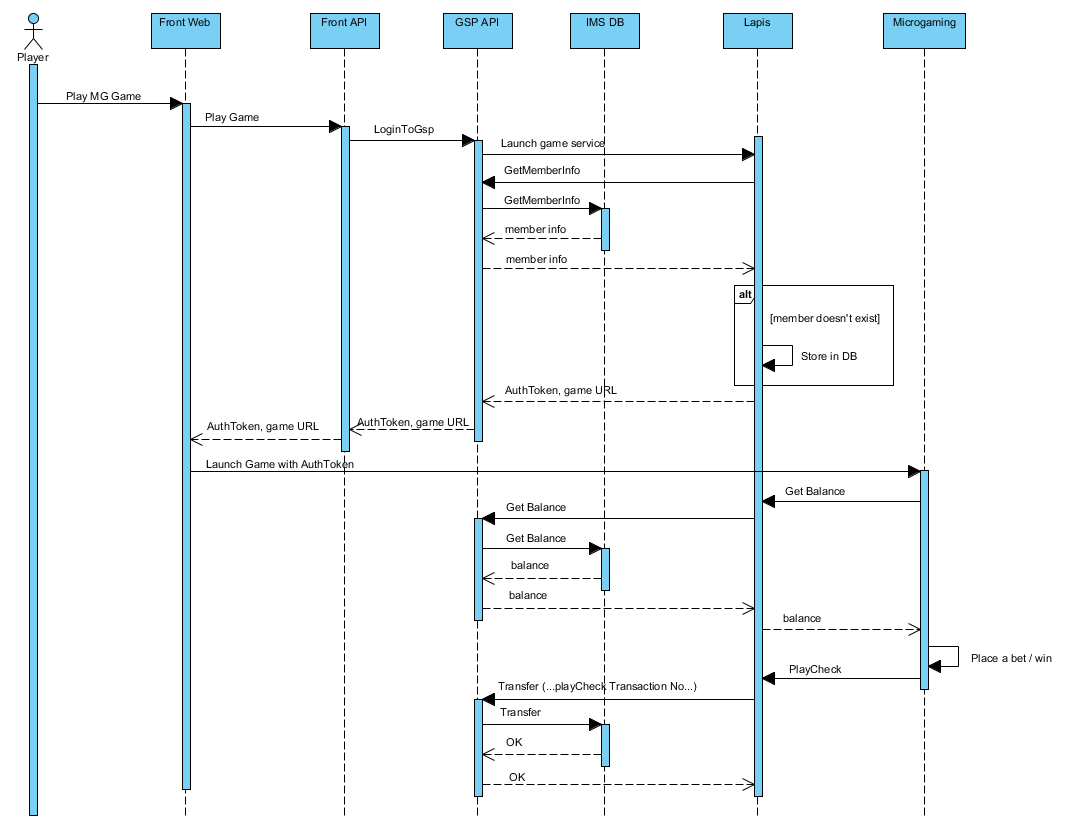
Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
|  |

* 1. Microgaming



|  |  |  |  |
| --- | --- | --- | --- |
| **Reference Document** | **Version** | **File Name** | **API** |
| Lapis - Single Wallet API - SWA | 0.01 | 0007-Lapis-SingleWalletAPI.pdf | Every API |

Lapis: A company that provides a service that connects between the IMS and microgaming

* + 1. LaunchGameService (POST)

**Mount Point: https://${host}/api-web/swa/api/${CODE}**

The API is used to launch games

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| seq | mandatory | string(50) | A unique id (usually it is advisable to generate using RANDAM\_UUID) for each call. It wasn’t used exactly anywhere, but this is good for auditing purposes. |
| token | mandatory | string(50) | Token is a unique identifier that will be used directly to be sent across to the integrating partner API. Token will be generated for each response to Lapis system, and in return lapis system will send the token to vendor system.  Token is tag with expiry time, this is to ensure  that the token wasn’t abuse. Partner is advised that the token should not be kept longer than 1 day. |
| lpsGameId | mandatory |  | the game id that wishes to launch |
| langCode | mandatory |  | language code for the game, if not supported, it will always be English  Web Browser Language Identfication Codes |
| currencyCode | mandatory | string | The member currency, current support currency code: USD  SGD  CNY  THB  VND  IDR  KRW  JPY  TWD  EUR  GBP  INR  MYR  ZAR |
| passPhrase | mandatory | string(50) | Each partner can generate password in lapis system in which we will be using it to communicate and to validate against each other. When passPhrase is changed on the Lapis system, it is advised that the Partner system is handling 2 password (old and new) until the old one has been entirely phased out |
| partnerId | mandatory | string | gsp id |

Example

|  |
| --- |
| <?xml version="1.0"?>  <launch-req seq="d940a25e-49f2-4d10-8e49-710d667300ed"  token="827d7387-d0c9-4154-b515-1ab59d3bc9c7"  passPhrase="demo-passphrase-HO"  lpsGameId="2"  langCode="en"  currencyCode="USD"  partnerId="HOR"  /> |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| status | mandatory | string(50) | status of the call |
| statusDesc | mandatory | string(255) | Description of the status |
| url | mandatory |  | a full url to the game pages (not SWF) that will launch the game |
| params | mandatory | property | any other param required to launch the game |

property

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| param name | mandatory | string |  |
| value | mandatory | string |  |

Example

|  |
| --- |
| <launch-resp token="" seq="" status="0" statusDesc="Ok" url="http://localhost:38080/simulator/game/test">  <params>  <param name="theme" value="158poker"/>  <param name="variant" value="vanguard"/>  <param name="AuthToken" value="LPS:3:1002:1004:2:@:3ac6fd20-25b3-11e3-93f2-d4bed9c2d64b"/>  <param name="gameid" value="MGS-SuperCarRacing"/>  <param name="ul" value="EN"/>  </params>  </launch-resp> |

|  |
| --- |
| <?xml version=”1.0”?>  <launch-resp seq=”” status=”999” statusDesc=”error” /> |

* + 1. PlayerDetail (POST)

**Mount Point: https://${partner\_url}**

The API is used to get player’s detail info by calling from Lapis (Micro)

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| seq | mandatory | string | A unique id (usually it is advisable to generate using RANDAM\_UUID) for each call. It wasn’t used exactly anywhere, but this is good for auditing purposes. |
| passPhrase | mandatory | string | Each partner can generate password in lapis system in which we will be using it to communicate and to validate against each other. When passPhrase is changed on the Lapis system, it is advised that the Partner system is handling 2 password (old and new) until the old one has been entirely phased out |
| token | optional | string | Token is a unique identifier that will be used directly to be sent across to the integrating partner API. Token will be generated for each response to Lapis system, and in return lapis system will send the token to vendor system.  Token is tag with expiry time, this is to ensure  that the token wasn’t abuse. Partner is advised that the token should not be kept longer than 1 day. |
| userId | mandatory | string | gspMember id |

Example

|  |
| --- |
| <?xml version="1.0"?>  <player-detail-req token="ef487460-ab08-4a30-aa35-acc9895f8083"  seq="f6410d52-48d7-40aa-98af-ec97d75f35eb"  passPhrase="demo-passphrase-HO"  userId=""/> |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| status | mandatory | string(50) | status of the call |
| statusDesc | mandatory | string(255) | Description of the status |
| token | mandatory | string | Token is a unique identifier that will be used directly to be sent across to the integrating partner API. Token will be generated for each response to Lapis system, and in return lapis system will send the token to vendor system.  Token is tag with expiry time, this is to ensure  that the token wasn’t abuse. Partner is advised that the token should not be kept longer than 1 day. |
| url | mandatory |  | a full url to the game pages (not SWF) that will launch the game |
| params | mandatory | Params | any other param required to launch the game |

params

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| param name | mandatory | string |  |
| value | mandatory | string |  |

Example

|  |
| --- |
| <?xml version="1.0"?>  <player-detail-resp seq="ef487460-ab08-4a30-aa35-acc9895f8083"  token="f6e080f3-fec3-44fb-9f2e-6c26d7ca8d55"  status="0"  statusDesc="Ok"  username="SWA-USD"  userId="1000"  firstName="Firstname - -USD"  lastName="Lasttname - SWA -USD"  country="SG"  currencyCode="USD"  balance="1000000"/> |

|  |
| --- |
| <?xml version=”1.0”?>  <launch-resp seq=”” status=”999” statusDesc=”error” /> |

* + 1. CreateMember

gspMember account is created automatically when Lapis receive a player’s detail information which is not stored in Lapis DB.

* + 1. PlayerBalance (POST)

**Mount Point:** https://${partner\_url}

This api is used to retrieve a member’s balance.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| seq | mandatory | string | A unique id (usually it is advisable to generate using RANDAM\_UUID) for each call. It wasn’t used exactly anywhere, but this is good for auditing purposes. |
| token | optional | string | Token is a unique identifier that will be used directly to be sent across to the integrating partner API. Token will be generated for each response to Lapis system, and in return lapis system will send the token to vendor system.  Token is tag with expiry time, this is to ensure  that the token wasn’t abuse. Partner is advised that the token should not be kept longer than 1 day. |
| passphrase | mandatory | string | Each partner can generate password in lapis system in which we will be using it to communicate and to validate against each other. When passPhrase is changed on the Lapis system, it is advised that the Partner system is handling 2 password (old and new) until the old one has been entirely phased out |
| userId | mandatory | string | gspMember id |

Example

|  |
| --- |
| <?xml version=”1.0”?>  <balance-req token="827d7387-d0c9-4154-b515-1ab59d3bc9c7"  seq="bd13d6eb-9b15-4afd-afaf-dd91b0ad2b35"  passPhrase="demo-passphrase-HO"  userId="1000"  /> |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| seq | mandatory | string | A unique id (usually it is advisable to generate using RANDAM\_UUID) for each call. It wasn’t used exactly anywhere, but this is good for auditing purposes. |
| token | mandatory | string | Token is a unique identifier that will be used directly to be sent across to the integrating partner API. Token will be generated for each response to Lapis system, and in return lapis system will send the token to vendor system.  Token is tag with expiry time, this is to ensure  that the token wasn’t abuse. Partner is advised that the token should not be kept longer than 1 day. |
| status | mandatory | string(50) | status of the call |
| statusDesc | mandatory | string(255) | Description of the status |
| balance | mandatory | long | Balance, in cents, e.g. if $1, this will be 100 |

Example

|  |
| --- |
| <?xml version="1.0"?>  <balance-resp seq="827d7387-d0c9-4154-b515-1ab59d3bc9c7"  token="0683458a-1061-4b39-ab0d-60b0e7bd233d"  status="0"  statusDesc="Ok"  balance="1000000"  /> |

|  |
| --- |
| <?xml version=”1.0”?>  <balance-resp seq=”827d7387-d0c9-4154-b515-1ab59d3bc9c7”  token=”0683458a-1061-4b39-ab0d-60b0e7bd233d”  status=”9”  statusDesc=”error”  /> |

* + 1. Transaction (POST)

**Mount Point:** https://${partner\_url}

The API is used to execute a transaction from Lapis when player places a bet or win the game.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| seq | mandatory | string |  |
| token | optional | string |  |
| passphrase | mandatory | string |  |
| userId | mandatory | string | gspMember Id  Normally when reconciliation took place, the token of the member is either already deleted by partner or already set to expire from partner.  In this case partner should detect that if token is not present or token is no-longer valid, partner should use userId to identify the member.  It is not wise to make use of userId to directly identify member. It is advisable if token is not available or token records no longer exist / expired then userId is used.  Partner should log necessary “balance changed” api called that doesn’t had valid token, for internal compliance check. |
| product | mandatory | string | product to be perform transaction  Product (casino/poker) |
| txType | mandatory | string | Transaction Type:  bet/win/refund -> common type for casino transfertovendor/transferfromvendor/refund -> common type for poker table game tournamentpurchase/transferfromvendor/win/refund -> common type for tournament game |
| lpsGameId | mandatory | string(50) | GameId -> representing “GAME” in Lapis System, e.g. Baccarat etc. |
| lpsTxId | mandatory | string(50) | Lapis internal TX ID for this call |
| vendorTxNo | optional | string(50) | on casino is refer to the playcheck Transaction No, however on poker this field is obsolete |
| vendorTxId | mandatory | string(50) | External (e.g MGS) tx id |
| pokerType | optional | string(50) | table / tournament |
| pokerId | optional | string(50) | when is poker product, this will hod either table or tournament id |
| tourmanetType | optional | string(50) | mtt / stt |
| lpsRefTxId | optional | string(50) | In case of Refund, reference tx Id will be given |
| amount | mandatory | long | Amount in cent |

Example

|  |
| --- |
| <?xml version=”1.0”?>  <transaction-req token="827d7387-d0c9-4154-b515-1ab59d3bc9c7"  seq="36c2893a-81f4-4765-83f8-8966a4480817"  passPhrase="demo-passphrase-HO"  userId="1000"  product="casino"  txType="bet"  lpsGameId="2"  lpsTxId="26"  vendorTxNo="1380093926511"  vendorTxId="13800939265111"  pokerType=""  pokerId=""  tournamentType=""  lpsRefTxId=""  amount="10000"  /> |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| seq | mandatory | string |  |
| token | mandatory | string |  |
| status | mandatory | string |  |
| statusDesc | mandatory | string |  |
| balance | mandatory | long | Balance, in cents, e.g. if $1, this will be 100  This is the balance after the transaction is applied |
| partnerTxId | mandatory | string(50) | Partner System TX ID that representing this tx call |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xml version="1.0"?>  <transaction-resp seq="827d7387-d0c9-4154-b515-1ab59d3bc9c7"  token="0d2081b7-0461-4236-996e-7591c7e460a7"  status="0"  statusDesc="Ok"  balance="1000000"  partnerTxId="1001"  /> |

|  |
| --- |
| <?xml version="1.0"?>  <transaction-resp seq="827d7387-d0c9-4154-b515-1ab59d3bc9c7"  token="0d2081b7-0461-4236-996e-7591c7e460a7"  status="9"  statusDesc="error"/> |

* + 1. Replay (what for?)

**Mount Point: https://${host}/api-web/swa/api/${CODE}/replay**

The API is used to ??

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| seq | mandatory | string | A unique id (usually it is advisable to generate using RANDAM\_UUID) for each call. It wasn’t used exactly anywhere, but this is good for auditing purposes. |
| token | optional | string | Token is a unique identifier that will be used directly to be sent across to the integrating partner API. Token will be generated for each response to Lapis system, and in return lapis system will send the token to vendor system.  Token is tag with expiry time, this is to ensure  that the token wasn’t abuse. Partner is advised that the token should not be kept longer than 1 day. |
| passPhrase | mandatory | string | Each partner can generate password in lapis system in which we will be using it to communicate and to validate against each other. When passPhrase is changed on the Lapis system, it is advised that the Partner system is handling 2 password (old and new) until the old one has been entirely phased out |
| langCode | mandatory |  | language code for the game, if not supported, it will always be english |
| currencyCode | mandatory | string | The member currency, current support currency code: USD  SGD  CNY  THB  VND  IDR  KRW  JPY  TWD  EUR  GBP  INR  MYR  ZAR |
| partnerId | mandatory | string | gsp id |

Example

|  |
| --- |
| <?xml version=”1.0”?>  <reply-req seq=”seq”  token=”token”  passPhrase=” passPhrase”  langCode=”en”  currencyCode=”USD”  partnerId=”HOR”  /> |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| status | mandatory | string(50) | status of the call |
| statusDesc | mandatory | string(255) | Description of the status |
| url | mandatory |  | a full url to the game pages (not SWF) that will launch the game |
| params | mandatory | params | any other param required to launch the game |

params

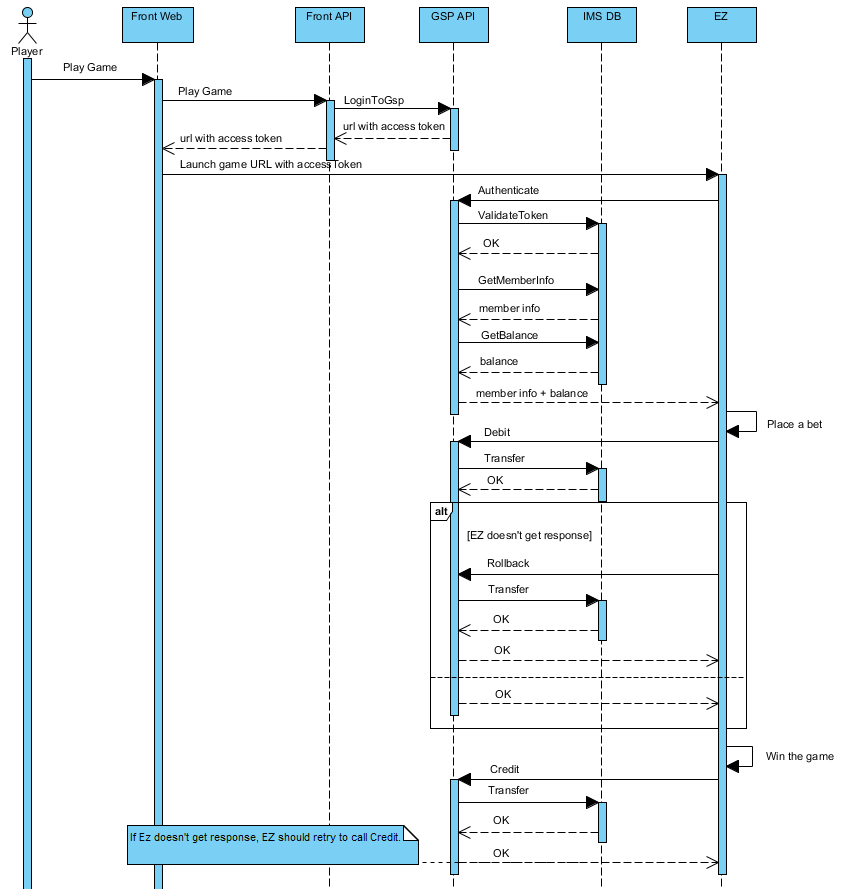
|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| param name | mandatory | string |  |
| value | mandatory | string |  |

Example

|  |
| --- |
| <?xml version=”1.0”?>  <reply-resp seq=”seq”  status=”0”  statusDesc=”abc”  url=”http://gm.vendor.com/gm/path/history.aspx”>  <param name=”param\_one”>  <![CDATA[value\_one]]>  </param>  <param name=”param\_two”>  <![CDATA[value\_two]]>  </param>  </reply-resp> |

|  |
| --- |
| <?xml version=”1.0”?>  <launch-resp seq=””  status=”999”  statusDesc=”error”  /> |

* 1. Ezuigi (EZ)



|  |  |  |  |
| --- | --- | --- | --- |
| **Reference Document** | **Version** | **File Name** | **API** |
| Live Casino API | v3.0.10 | Live casino API-v.3.0.10.pdf |  |

* + 1. PlayGame (POST) – Open Game Lobby

Mount Point: ~/game

The API is used to call before a player launch the game lobby.

For the time being, it is recommended to only use English.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| operatorId | mandatory | int(11) | gsp id?  Defines the particular operator that the player belongs to.  This is allocated by the Live Casino system in advance. |
| token | mandatory | varchar(250) | Initial token generated by the operator based upon the username and password. Will be used "as is" in order to authenticate the user. |
| language | mandatory | char(2) | Specifies the desired language file to load for the current game.  Default English (en) : ISO 639-1 |
| clientType | mandatory | varchar | Flash (In the future it will also be available in HTML5) |
| viewport | optional | varchar | 1. Changes to the lobby skin (nano, headerless)  2. Changes to the way a game opens- can be configured to open in the same window or a separate one. |
| postMessage | optional | int(1) | When using postMessage, navigating between games is disabled.  Instead a JSON is sent to the opening page (operator parent page or opener window) containing the table id and the user selected.  example:  {"msg\_type":"open\_table", "game": "blackjack", "table\_id":1 }  This is useful when the operator wants to control the page opening for the game |
| wmode | optional | ??? | can be direct or transparent |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameURL | mandatory | string | game url? |

Example

|  |
| --- |
| https://lobbyqa.ezugi.com/common/operator\_proxy.php? language=en&operatorId={generatedByLiveCasinoSide}&token={generatedValue}&clientType={valuefromlist(html5,flash)},'Lobby','height=650, width=960, resizable=no, scrollbars=no |

* + 1. Authentication (POST) - Authentication of User API

Mount Point: ~/game

The Authentication API enables the Live Casino system to verify the information received in the launch protocol. Only once the operator approves this information will the lobby open up to the player.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| operatorId | mandatory | int(11) | Defines the particular operator that the player belongs to. |
| brandId | optional | int(11) | Do not use anymore. Will be removed in the near future. |
| token | mandatory | varchar(250) | access token  Initial player’s token to launch the protocol.  This will be returned as the playersTokenAtLaunch. |

Example

|  |
| --- |
| {  "operatorId":11000011,  "token":"2e50335534e3af94959b1abaac1ce789",  "brandId":0  } |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| operatorId | mandatory | int(11) | The operator’s unique identifier. |
| uid | mandatory | varchar(50) | gspMember id  An operator’s unique identification for each player. |
| nickName | mandatory | varchar(100) | gspMember id  characters and numbers only |
| token | mandatory | varchar(250) | access token  Not to be confused with the input token. This token is the session token- A unique identifier which is generated by the operator to identify a session’s interactions. |
| playerTokenAtLaunch | madantory | varchar(250) | access token  The player’s initial token from operator (by launching the lobby) – we use this for validation. |
| balance | mandatory | double(25,2) | player’s balance  Available money in the player’s wallet at the time of the request. The value will be presented in the currency of that session. |
| currency | mandatory | char(3) | player currency ISO code |
| language | mandatory | char(2) | Specifies the desired language file to load for the current game.  We will support both 2-3 ISO code Lang type.  Default English. |
| date | mandatory | datetime | The date and time of the request as it appears in the Live Casino server.  format: yyyy-mm-dd hh:mm:ss:mss or  yyyymmddhhmmssmsss |
| clientIP | manatory | varchar(30) | IP address in Login() can be used.  The Internet IP of the computer on which the end user is playing |
| errorcode | mandatory | tinyint(4) | Identifies whether or not the request has been processed successfully.  If no error is detected, this code value is 0. |
| errorDescription | mandatory | varchar(100) | A string that describes the response.  This string is not the message to the player but rather gives details of the error. |
| timestamp | ?? | string | Time representation in UNIX milliseconds format.  To see the system time in Ezugi system you can check  http://engine.ezugi.com:8080/GameServer/OperatorGateController?action=getTimeInMillis |

Example

|  |
| --- |
| {  "operatorId":11000011,  "UID":"A0000000008",  "nickName":"ivashka",  "token":"ae5fc2490afe4a1d0da4d2c06f821dbd",  "playerTokenAtLaunch":"2e50335534e3af94959b1abaac1ce789",  "balance":16410,  "currency":"USD",  "language":"en",  "date":"2013-12-30 11:15:43:156",  "clientIP":"127.0.0.1",  "errorCode":0,  "errorDescription":"ok",  } |

* + 1. Debit (POST)

Mount Point: ~/chips

The Debit API enables the Live Casino platform to withdraw money from the player’s wallet through the operator’s site.

This financial transaction will be done in three cases:

1. When a regular bet is placed,
2. When a regular and a side bet are placed
3. When the dealer is tipped.

In the event of a final error response- The Live Casino server will NOT repeat the call. A relevant message is presented to the user.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| serverId | mandatory | int(11) | The unique serverId of the Live Casino system. |
| opeartorId | mandatory | int(11) | gsp id  Defines the particular operator that the player belongs to. |
| brandId | optional | int(11) | Do not use anymore.  Will be removed in the near future. |
| token | mandatory | varchar(250) | A unique identifier which is generated and sent from the operator to a running Live Casino, in order to identify an interaction session. |
| uid | mandatory | varchar(50) | gspMember id  An operator’s unique identification for each player. |
| date | mandatory | datetime | The date and time in the operator’s server when a transaction leaves. |
| transactionId | mandatory | char(50) | A unique key to indicate a specific financial activity. This key will guarantee that a transaction is handled only once. |
| roundId | mandatory | bigint(18) | A unique key per server ID, to identify a game cycle. Games with more than one financial cycle will have the same roundId for all requests. |
| financialId | optional | bigint(18) | Do not use anymore. Will be removed in the near future. |
| gameId | mandatory | tinyint(4) | The lobby will have a separate gameId assigned in order to open it. |
| currency | mandatory | char(3) | player currency ISO code |
| debitAmount | mandatory | double | The amount requested to be withdrawn from the player’s wallet.  The value must be a positive number above zero. |
| betTypeId | mandatory | tinyint | Debit Index Types  1 Table bet  2 Reserved for future use  3 Tip \*  4 BJ insurance  5 BJ double  6 BJ split  7 BJ ante  8 BJ jackpot  9 Poker ante  10 Poker wager  11 Poker jackpot  12 Sidebets 1   1. Sidebets 2   Tip: The tip bet is a special type of debit that is not a part of the game’s financial actions (bets and "win/lose"). This parameter is used for correct tip handling. It does not start or end any round and could take place in between rounds as well. When playing blackjack, the player may only tip once he has a seat in order for it to be clear who the tip belongs to. |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| operatorId | mandatory | int(11) | gsp id  The operator’s unique identifier |
| roundId | mandatory | bigint(18) | A unique key per server ID, to identify a game cycle.  Games with more than one financial cycle will have the same roundId for all requests. |
| uid | mandatory | varchar(50) | gspMember id  An operator’s unique identification for each player. |
| token | mandatory | varchar(50) | access token  A unique identifier which is generated and sent from the operator to a running Live Casino, in order to identify an interaction session |
| balance | mandatory | double(5,2) | Available money in the player wallet at the time of the request.  The value will be presented in the currency selected for the session |
| transactionId | mandatory | char(50) | transaction id  A unique key to indicate a specific financial transaction. This key will guarantee that a transaction is handled only once. |
| currency | mandatory | char(3) | player currency ISO code |
| date | mandatory | string | The date and time of when the response transaction leaves; as it appears in the operator's server.  format: yyyy-mm-dd hh:mm:ss:mss or yyyymmddhhmmssmsss |
| errorCode | mandatory | tinyint(4) | Identifies whether or not the request has been processed successfully.  If no error is detected, this code value is 0. |
| errorDescription | mandatory | varchar(100) | A string that describes the response.  This string is not the message to the player but rather details of the error. |
| timestamp | mandatory | string | Time representation in UNIX milliseconds format.  To see the system time in Ezugi system you can check  http://engine.ezugi.com:8080/GameServer/OperatorGateController?action=getTimeInMillis |

Example

|  |
| --- |
| {  “tranid”:19389,  “status” : “approved”  } |

|  |
| --- |
| {  “error”: 4004,  “message”: “Player’s balance is insufficient to withdraw.”  “status” : “declined”  } |

* + 1. Rollback (POST)

Mount Point: ~/chips

This api is used to rollback a transaction.

1. No response is received from the operator when no more bets are accepted.

2. Operator sends an exception.

3. Error codes: 1, 2, 10 ( 4.4.6 Error codes )

|  |  |
| --- | --- |
| 1 | General error |
| 2 | Saved for future use |
| 10 | Transaction timed out |

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| serverId | mandatory | int(11) | The unique serverId of the Live Casino system. |
| opeartorId | mandatory | int(11) | gsp id  Defines the particular operator that the player belongs to. |
| brandId | optional | int(11) | Do not use anymore.  Will be removed in the near future. |
| token | mandatory | varchar(250) | The player’s token as received when the Live Casino was launched. |
| uid | mandatory | varchar(50) | gspMember id  An operator’s unique identification for each player. |
| date | mandatory | datetime | The date and time in the operator’s server when a transaction leaves. |
| transactionId | mandatory | char(50) | A unique key to indicate a specific financial activity. This key will guarantee that a transaction is handled only once. |
| roundId | mandatory | bigint(18) | A unique key per server ID, to identify a game cycle. Games with more than one financial cycle will have the same roundId for all requests. |
| financialId | optional | bigint(18) | Do not use anymore. Will be removed in the near future. |
| gameId | mandatory | tinyint(4) | The lobby will have a separate gameId assigned in order to open it. |
| currency | mandatory | char(3) | player currency ISO code |
| rollbackAmount | mandatory | double | The amount requested to be credited to the player’s wallet.  Value >=0. |
| seatId | mandatory | string | The seat index number from where the debit occurred (Relevant only to BJ)  “s1” to “s7” refers to the player’s initial seat.  “s1-2” to “s7-2” refers to split seats.  “s0” refers to tips. (A tip can be places without acquiring a seat.) |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| operatorId | mandatory | int(11) | gsp id  The operator’s unique identifier |
| roundId | mandatory | bigint(18) | A unique key per server ID, to identify a game cycle.  Games with more than one financial cycle will have the same roundId for all requests. |
| uid | mandatory | varchar(50) | gspMember id  An operator’s unique identification for each player. |
| token | mandatory | varchar(50) | access token  A unique identifier which is generated and sent from the operator to a running Live Casino, in order to identify an interaction session. |
| balance | mandatory | double(5,2) | Available money in the player wallet at the time of the request.  The value will be presented in the currency selected for the session |
| transactionId | mandatory | char(50) | transaction id  A unique key to indicate a specific financial transaction. This key will guarantee that a transaction is handled only once. |
| currency | mandatory | char(3) | player currency ISO code |
| date | mandatory | string | The date and time of when the response transaction leaves; as it appears in the operator's server.  format: yyyy-mm-dd hh:mm:ss:mss or yyyymmddhhmmssmsss |
| errorCode | mandatory | tinyint(4) | Identifies whether or not the request has been processed successfully.  If no error is detected, this code value is 0. |
| errorDescription | mandatory | varchar(100) | A string that describes the response.  This string is not the message to the player but rather details of the error. |
| timestamp | mandatory | string | Time representation in UNIX milliseconds format.  To see the system time in Ezugi system you can check  http://engine.ezugi.com:8080/GameServer/OperatorGateController?action=getTimeInMillis |

Example

|  |
| --- |
| {  “tranid”:19389,  “status” : “approved”  } |

|  |
| --- |
| {  “error”: 4005,  “message”: “External transaction id {0} is duplicated..”,  “status” : “declined”  } |

|  |  |
| --- | --- |
| **ERROR** | **MESSAGE** |
| 2003 | The player doesn’t exist |
| 4000 | The deposit limit for the ongoing period has been exceeded |
| 4001 | The agent has insufficient funds in his balance |
| 4002 | Amount exceeds maximum deposit limit |
| 4004 | Player’s balance is insufficient to withdraw |
| 4005 | External transaction id is duplicated |

* + 1. Credit (POST)

Mount Point: ~/chips

The Credit API enables the Live Casino platform to perform a financial transaction of returning money into the player's wallet through the operator’s server.

When calculating the results, a player that loses, has a credit amount of 0.

A player that wins, is credited the total of the payout amount.

Credit calls may occur in the middle of a game round in the following situations:

1. When a player is out of rounds and should be refunded for his accepted bet.

2. In the Blackjack game, if a player decides to "Surrender", he will immediately get refunded and kicked out of the round.

In the event of a final error response- The Live Casino server will NOT repeat the call.

A relevant message is presented to the user.

 In the event of:

3. Server Error- Any potential error inside the operator's system excluding an HTTP exception type, such as 400, 402, etc.

4. Socket Timeout- which may occur when the connection is closed by the system.

The Live Casino server will repeat the Credit call three times in ten second intervals.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| serverId | mandatory | int(11) | The unique serverId of the Live Casino system. |
| opeartorId | mandatory | int(11) | gsp id  Defines the particular operator that the player belongs to. |
| brandId | optional | int(11) | Do not use anymore.  Will be removed in the near future. |
| token | mandatory | varchar(250) | The player’s token as received when the Live Casino was launched. |
| uid | mandatory | varchar(50) | gspMember id  An operator’s unique identification for each player. |
| date | mandatory | datetime | The date and time in the operator’s server when a transaction leaves. |
| transactionId | mandatory | char(50) | A unique key to indicate a specific financial activity. This key will guarantee that a transaction is handled only once. |
| roundId | mandatory | bigint(18) | A unique key per server ID, to identify a game cycle. Games with more than one financial cycle will have the same roundId for all requests. |
| financialId | optional | bigint(18) | Do not use anymore. Will be removed in the near future. |
| gameId | mandatory | tinyint(4) | The lobby will have a separate gameId assigned in order to open it. |
| currency | mandatory | char(3) | player currency ISO code |
| creditAmount | mandatory | double | The amount requested to be credited to the player’s wallet.  Value >=0. |
| returnReason | mandatory | tinyint(4) | Defines the reason for returning the money.  0 Successful bet  1 Cancel bet  2 Canceled round  7 Manual Credit |
| isEndRound | mandatory | tinyint(4) | Defines whether this financial transaction is the final transaction of the specific round.  Will contain yes/no parameters.  \* Here Tinyint came as Boolean field type  because MySql doesn't contain Boolean type |
| gameDataString | optional | varchar(1000) | only in end round  An XML with the game data history of only the specific player, public bets, moves and results.  This will be sent only at the end of a round. |
| seatId | mandatory | string | The seat index number from where the debit occurred (Relevant only to BJ)  “s1” to “s7” refers to the player’s initial seat.  “s1-2” to “s7-2” refers to split seats.  “s0” refers to tips. (A tip can be places without acquiring a seat.) |
| betTypeId | mandatory | string | Credit Index Types  0  108 BJ jackpot win  111 Poker jackpot  112 Ezugi sidebets payout  113 Third party licensed side bets payout |

Example

|  |
| --- |
| {  "uid":"A0000000008",  "financialId":1169445,  "transactionId":"30515bee-7833-4d7a-a40a-7b63c8e79e1b",  "operatorId":11000011,  "gameId":3,  "isEndRound":true,  "brandId":0,  "currency":"USD",  "gameDataString":"<?xmlversion='1.0'encoding='UTF8'?>  <GameString>  <gameData>  <roundID>1169445<\/roundID>  <gameType>Roulette<\/gameType>  <tableid>1001…  GameString>",  "token":"ae5fc2490afe4a1d0da4d2c06f821dbd",  "creditAmount":540.0,  "serverId":1,  "roundId":1169445,  "returnReason":0  } |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| operatorId | mandatory | int(11) | gsp id  The operator’s unique identifier |
| roundId | mandatory | bigint(18) | A unique key per server ID, to identify a game cycle.  Games with more than one financial cycle will have the same roundId for all requests. |
| uid | mandatory | varchar(50) | gspMember id  An operator’s unique identification for each player. |
| token | mandatory | varchar(50) | access token  A unique identifier which is generated and sent from the operator to a running Live Casino, in order to identify an interaction session. |
| balance | mandatory | double(5,2) | Available money in the player wallet at the time of the request.  The value will be presented in the currency selected for the session |
| transactionId | mandatory | char(50) | transaction id  A unique key to indicate a specific financial transaction. This key will guarantee that a transaction is handled only once. |
| currency | mandatory | char(3) | player currency ISO code |
| date | mandatory | string | The date and time of when the response transaction leaves; as it appears in the operator's server.  format: yyyy-mm-dd hh:mm:ss:mss or yyyymmddhhmmssmsss |
| errorCode | mandatory | tinyint(4) | Identifies whether or not the request has been processed successfully.  If no error is detected, this code value is 0. |
| errorDescription | mandatory | varchar(100) | A string that describes the response.  This string is not the message to the player but rather details of the error. |
| timestamp | mandatory | string | Time representation in UNIX milliseconds format.  To see the system time in Ezugi system you can check  http://engine.ezugi.com:8080/GameServer/OperatorGateController?action=getTimeInMillis |

Example

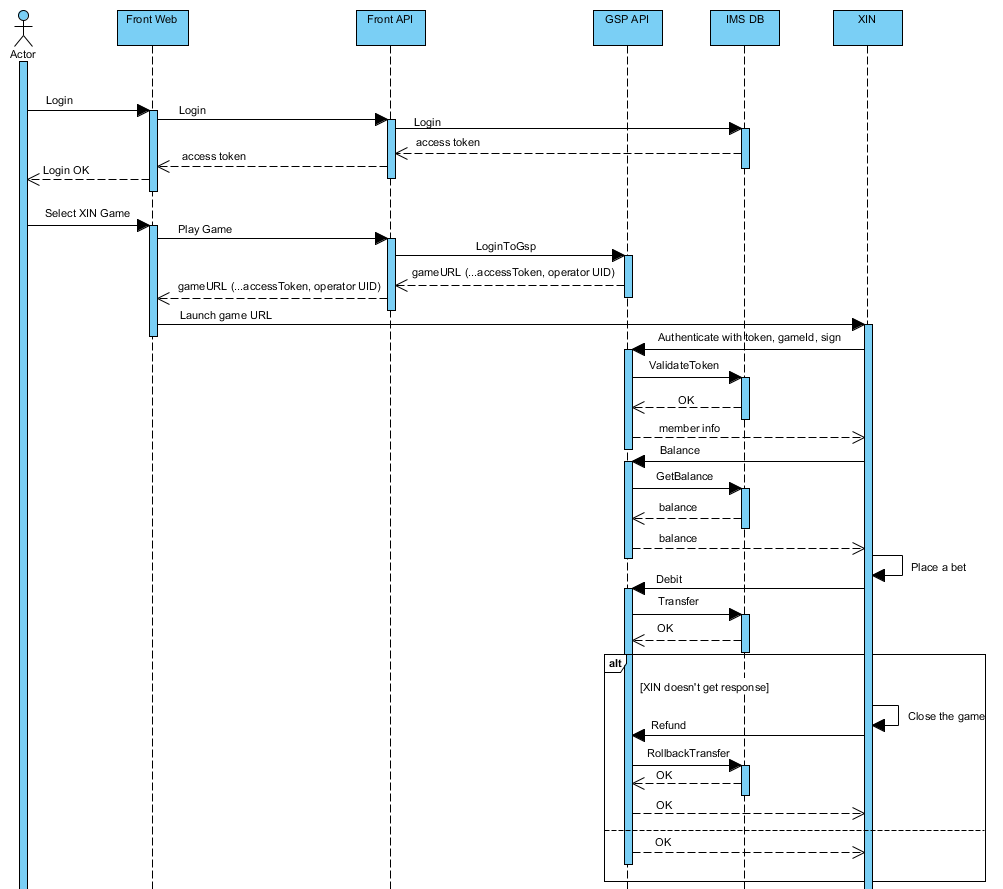
|  |
| --- |
|  |

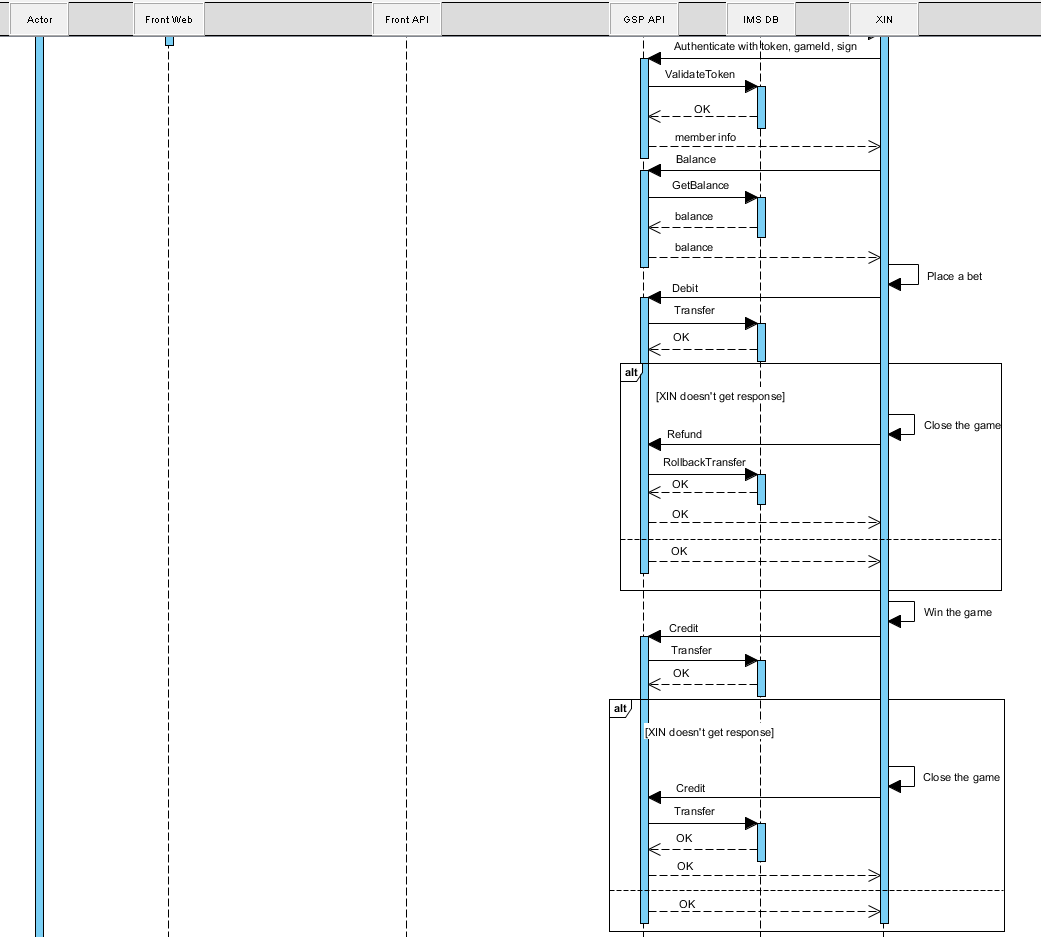
|  |
| --- |
|  |

* + 1. Error Codes

|  |  |
| --- | --- |
| **Error ID** | **Error Description** |
| 0 | Completed successfully |
| 1 | General error |
| 2 | Saved for future use |
| 3 | Insufficient funds |
| 4 | Operator limit to the player 1 (insufficient behavior) |
| 5 | Operator limit to the player 2 (insufficient behavior) |
| 6 | Token not found |
| 7 | User not found |
| 8 | User blocked |
| 9 | Transaction not found |
| 10 | Transaction timed out |
| 11 | Reserved for the future use |

* 1. XIN Gaming (XIN)





|  |  |  |  |
| --- | --- | --- | --- |
| **Reference Document** | **Version** | **File Name** | **API** |
| **Game Integration API** | **V 1.2** | **XIN Gaming - Game Integration API.pdf** |  |

* + 1. Launching a Game

Mount Point: ~/?token={token}&operator={operatorUid}

The API is used to call before a player launch the game lobby.

For the time being, it is recommended to only use English.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token |
| operatorUid | mandatory | string | the unique operator UID  (allocated to the operator by XIN Gaming) |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameURL | mandatory | string | game URL |

Example

|  |
| --- |
|  |

* + 1. Authentice (POST)

**Mount Point**: ~/**authenticate**

When a game is launched, the XIN Gaming Remote Gaming Server(XRGS) will attempt to authenticate the member with the operator via this API.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| token | mandatory | varchar(40) | access token |
| game\_id | mandatory | varchar(32) | XIN game id |
| sign | mandatory | varchar(40) | request signature  please review sample  1. make a string in parameter name alphabetical order  2. add the secret key  3. SHA1 encryption  IMS must then validate the request by checking this signature. |

Example

|  |
| --- |
|  |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| player\_id | mandatory | varchar(32) | gspMember id |
| currency | mandatory | varchar(3) | currency ISO code |
| language | optional | varchar(2) | language code : ISO 639-1 ? |
| nickname | optional | varchar(32) | gspMember id |

Example

|  |
| --- |
| {  “player\_id” : “123456789”  “currency” : “GBP”,  “language” : “en”,  “nickname” : “FrankyStein”  } |

* + 1. Debit (POST)

**Mount Point**: ~/**debit**

This method is called whenever the gspMember makes a bet and their balance is to be debited.

Each debit transaction has a unique XRGS identifier (transaction\_id).

The operator must check this identifier to ensure that a transaction from the XRGS is not processed twice.

However, if a duplicate identifier is sent to the API, the operator must return the previous response (without affecting the players balance) and pass the ‘already\_processed’ flag as ‘true’.

If a /debit call times out, the XRGS server will force the game to shut down and then attempt to refund the bet via the /refund API.

It will attempt the refund multiple times (configurable in the XRGS) until success or if the retry limit has been reached.

(Please refer to the Communication Flow – Timeouts diagram.)

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| token | mandatory | varchar(40) | access token |
| game\_id | mandatory | varchar(32) | XIN game id |
| sign | mandatory | varchar(40) | request signature  please review sample  1. make a string in parameter name alphabetical order  2. add the secret key  3. SHA1 encryption  IMS must then validate the request by checking this signature. |
| transaction\_id | mandatory | int | transaction id from XIN |
| id | mandatory | int | unique bet/ game round id |
| amount | mandatory | int | the debit amount in cent  e.g. 15.00 USD is represented as 1500 |
| date | mandatory | int | date of transaction (unix timestamp) |
| decription | mandatory | varchar(32) | The description of the debit/bet.  Currently the only value for this will be ‘BET’. |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| transaction\_id | mandatory | varchar(32) | processed transaction id in IMS |
| balance | mandatory | int | player balance after transaction (in cent) |
| already\_processed | mandatory | Boolean | true if the requested transaction Id has been processed previously |

Example

|  |
| --- |
| {  “transaction\_id” : “54321”,  “balance” : 95000,  “already\_processed” : false  } |

* + 1. Credit (POST)

**Mount Point**: ~/**credit**

This method is called whenever the XRGS needs to credit the player’s balance (i.e. a win)

Similarly, if a /credit call times out, the XRGS will force the game to shut down and then attempt the /credit call again until success or the retry limit has been reached (again, configurable in the XRGS)

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| token | mandatory | varchar(40) | access token |
| game\_id | mandatory | varchar(40) | XIN game id |
| sign | mandatory | varchar(40) | request signature  please review sample  1. make a string in parameter name alphabetical order  2. add the secret key  3. SHA1 encryption  IMS must then validate the request by checking this signature. |
| transaction\_id | mandatory | int | transaction id from XIN |
| id | mandatory | int | the unique bet/ round id |
| amount | mandatory | int | the credit amount in cent |
| date | mandatory | int | date of transaction (UNIX timestamp) |
| description | mandatory | varchar(32) | The description of the credit/win.  Current values are ‘WIN’, ‘FEATURE:XXXX’,  PROGRESSIVE:XXXX’ (where XXXX is the  name of the feature or progressive win) |

Example

|  |
| --- |
|  |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| transaction\_id | mandatory | varchar(32) | processed transaction id in IMS |
| balance | mandatory | int | player balance after transaction |
| already\_processed | mandatory | boolean | True if the requested transaction Id has been processed previously |

Example

|  |
| --- |
| {  “transaction\_id” : “54321”,  “balance” : 105000,  “already\_processed” : false  } |

* + 1. Balance (POST)

**Mount Point**: ~/**balance**

This method is used to get the players balance.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| token | mandatory | varchar(40) | access token |
| sign | mandatory | varchar(40) | request signature  please review sample  1. make a string in parameter name alphabetical order  2. add the secret key  3. SHA1 encryption  IMS must then validate the request by checking this signature. |

Example

|  |
| --- |
|  |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| balance | mandatory | int | player balance in cent (100ths of money units) |

Example

|  |
| --- |
| {  “balance” : 80000  } |

* + 1. Refund (POST)

**Mount Point**: ~/**refund**

This method is called whenever a debit amount needs to be returned to the players balance.

The request parameters will match the exact parameter values that were sent in the debit call we wish to refund.

The operator should cancel any previous debit with the matching XRGS transaction Id and return the debit amount to the players balance.

If the previous debit with the specified transaction\_id has not yet been processed, the operator must record this refund data and NOT process any subsequent debits with this transaction\_id.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| token | mandatory | varchar(40) | access token |
| game\_id | mandatory | varchar(40) | XIN game id |
| sign | mandatory | varchar(40) | request signature  please review sample  1. make a string in parameter name alphabetical order  2. add the secret key  3. SHA1 encryption  IMS must then validate the request by checking this signature. |
| transaction\_id | mandatory | int | transaction id from XIN |
| id | mandatory | int | the unique bet/ round id |
| amount | mandatory | int | the debit amount in cent |
| date | mandatory | int | date of transaction (UNIX timestamp) |

Example

|  |
| --- |
|  |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| transaction\_id | mandatory | varchar(32) | processed transaction id in IMS |
| balance | mandatory | int | player balance after transaction |
| already\_processed | mandatory | boolean | True if the XRGS transaction Id has already been refunded before |

Example

|  |
| --- |
| {  “transaction\_id” : “54321”,  “balance” : 105000,  “already\_processed” : false  } |

* + 1. Error Handling

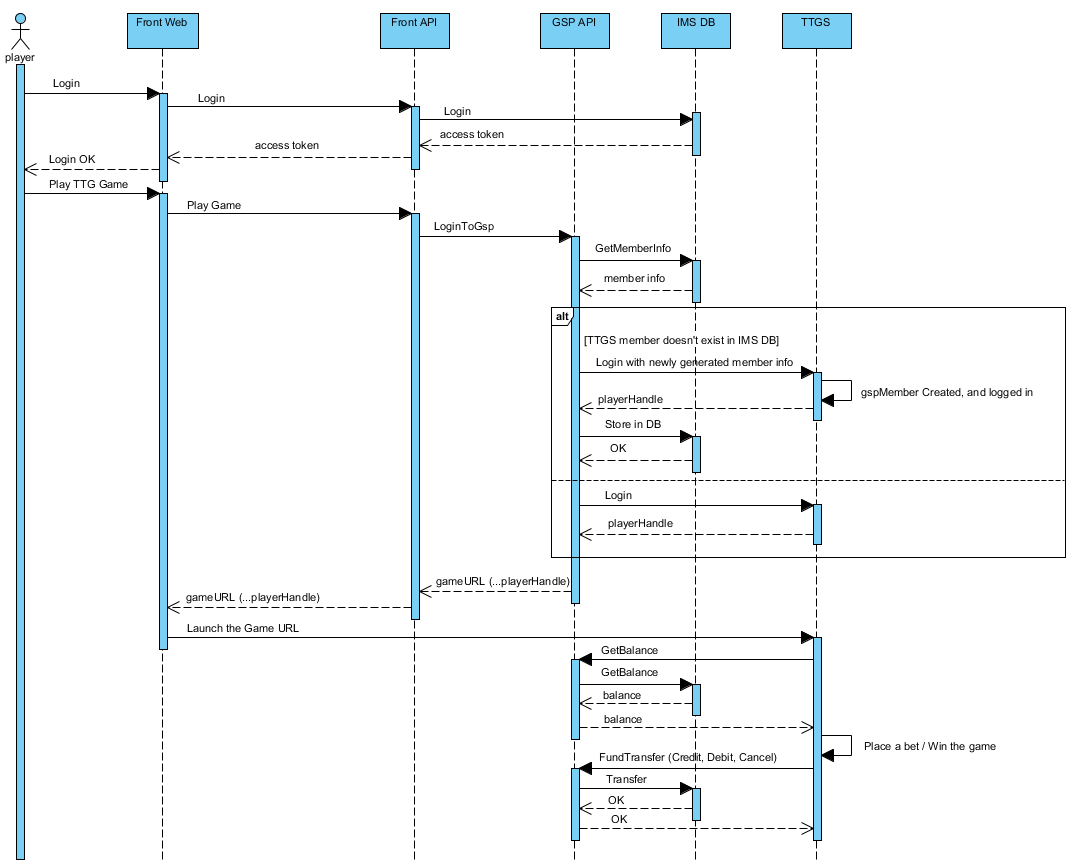
|  |  |  |
| --- | --- | --- |
| **Status Code** | **Code** | **Description** |
| 401 | ACCESS\_DENIED | There has been a problem validating the request signature |
| 402 | INSUFFICIENT\_FUNDS | /debit failed as the player doesn’t have the funds for the debit/bet |
| 403 | TOKEN\_EXPIRED | The operator has deemed that the access token has expired and the token is now invalid |
| 404 | TOKEN\_NOT\_FOUND | The token cannot be found |
| 500 | INTERNAL\_ERROR | There has been an internal error (unspecified) |

Example

HTTP Status: 402

|  |
| --- |
| {  “code” : “INSUFFICIENT\_FUNDS”,  “message” : “The player has insufficient funds”  } |

* 1. Top Trend Gaming (TTGS or Amaya)



|  |  |  |  |
| --- | --- | --- | --- |
| **Reference Document** | **Version** | **File Name** | **API** |
| **Integration Services API** | **2012 Q1** | **[사용중]Top Trend Integration Services API - 2012 Q1.pdf** | **ALL** |
| **Integration Services API** | **V 13.3** | **AMS Integration Services API - 13.3.pdf** | **updated on 2015-06-01** |

* + 1. Login

Mount Point: ~/? method=**login**&uid={**uid**}&account={**account**}&currency={**currency**} &country={**country**}&integrationtype={**integrationtype**}&sessionToken={**sessionToken**}&commonwallet={**commonwallet**}

The API is used to login TTGS.

If the gspMember does not exist in the TTGS initially, the gspMember is created, and logged in.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| method | mandatory | string | login |
| xmlresponse | mandatory | string | true |
| uid | mandatory | string(32) | gspMember id  NOTE: You must use a prefix (customer identifier) that will be given to you when the integration is started.  For example, site123\_playersID where site123 is the prefix that is supplied to you that represents your site in the gaming platform. |
| account | mandatory | string | parameter?  single wallet |
| currency | mandatory | string | currency ISO code |
| country | mandatory | string | gspMember country of residence  ISO 3166 alpha-2  Use xx in case of a FunMoney player |
| integrationtype | mandatory | string | 0: Chip Transfer type  2: Standard HTTP Common Wallet  ~~3: OpenBet Oxi Common Wallet~~ |
| sessionToken | mandatory | string | access token |
| commonwallet | mandatory | string | REQUIRED For type 1 common wallet only. |
| lsdId | optional | string | Specifies from whom the player is logging in from.  A hierarchy can be specified. Max 24 characters.  If you pass in non-existent lsdId value, the affiliate will be created on the fly as the partner type.  You can pass in an affiliate path and a hierarchy will be created.  zero: 0  Partner: 1  Webmaster:2  See Broker Support for SDK Login for details. |
| fn | optional | string | First name, (max. 50 characters), defaults to Guest if not specified.  Anything entered here will become the player's first name, replacing whatever was entered before. |
| ln | optional | string | Last name, (max. 50 characters), defaults to Guest if not specified.  Anything entered here will become the player's last name, replacing whatever was entered before. |
| un | optional | string | User name, (max. 50 characters), defaults to the unique identifier if not specified.  The user name must only consist of alpha-numeric characters. Must also be unique.  NOTE: You must use a prefix (customer identifier) that will be given to you when the integration is started.  For example, site123\_UserName where site123 is the prefix that is supplied to you that represents your site in the gaming platform. |
| ~~nn~~ | ~~optional~~ | ~~string~~ | ~~Nick name, (max. 50 characters), defaults to the unique identifier if not specified. Must also be unique.~~ |
| ~~bal~~ | ~~optional~~ | ~~string~~ | ~~Balance, (value must be greater than 0).~~  ~~Value represents the amount to be added to the user’s account.~~ |
| ~~regbonusplanid~~ | ~~optional~~ | ~~string~~ | ~~Registration bonus plan ID, optional parameter passed in if a specific registration bonus is to be used.~~  ~~Does not default to any value if it is not specified.~~ |
| ~~applyregbonus~~ | ~~optional~~ | ~~string~~ | ~~Can be passed in for login method to indicate whether or not a registration bonus should be applied.~~  ~~Can be set to either true or 1.~~  ~~Anything else means no bonus will be applied. regbonusplanid (above) is also optional for getting a registration bonus.~~  ~~a. If regbonusplanid is passed in and applyregbonus set to true, the specified bonus will be attempted;~~  ~~b. If regbonusplanid is passed in and applyregbonus set to false, no bonus will be applied as the flag overwrites;~~  ~~c. If regbonusplanid is not passed in and applyregbonus is set to true, the system will find a suitable registration bonus;~~  ~~d. If regbonusplanid is passed in and no applyregbonus found, the specified bonus will be attempted;~~  ~~Both applyregbonus and regbonusplanid are optional; if neither is found, no registration bonus will be applied.~~ |
| ~~depbonusplanid~~ | ~~optional~~ | ~~string~~ | ~~Deposit bonus plan ID must be passed in if deposit bonus is being used when passing in a player balance on login.~~  ~~Defaults to best available bonus plan that is set up in the casino system.~~  ~~If no bonus plan id or a bonus plan id of 0 is submitted, then the bonus plan with the highest priority is used.~~  ~~If an invalid bonus plan id is used (an integer greater than 0 that is not associated with a bonus plan), then no bonus is applied.~~  ~~However, if a negative bonus plan id is used (an integer less than 0) no bonus is applied since no bonus deposit event will be created.~~  ~~This is more efficient than using an invalid positive bonus plan id if you do not want a bonus.~~ |
| ~~broker~~ | ~~optional~~ | ~~string~~ | ~~Must be set to true if this player is to be created as a broker. Brokers are described in Broker Support for Login.~~ |
| ~~couponamt~~ | ~~optional~~ | ~~string~~ | ~~Number of coupons - must be an integer.~~  ~~Must be positive.~~ |
| tester | optional | string | 0: not a tester (defalut)  1: tester |
| ~~token~~ | ~~optional~~ | ~~string~~ | ~~This token will be required to be supplied, depending on the requirements of the common wallet.~~  ~~Currently support is only available for Common Wallet Type 2 and 3.~~ |

Example

|  |
| --- |
| https://casinoHost/sdk/servlet/com.chartwelltechnology.icd.ClientLoginSupportServlet?method=login&uid=1536&fn=Jill&ln=Atkins&nn=bigspender&un=biggerspender&account=GBP&country=DK& lsdId=zero |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameURL | mandatory | string | game URL |

|  |  |  |  |
| --- | --- | --- | --- |
| **ERROR CODE** | **DESCRIPTION** | **REASON CODE** | **DESCRIPTION** |
| 4 | Missing Required Fields. |  |  |
| 7 | Balance Entered Is Invalid. |  |  |
| 8 | Unable to Login. - see notes below for more details on this error code. | 1 | Unknown |
| 2 | Duplicate User |
| 3 | DB error |
| 4 | Blocked User |
| 11 | Account Entered Is Invalid. |  |  |
| 12 | Bonus Plan Id Is Not A Number. |  |  |
| 14 | Balance must be 0 when opening a common wallet account. |  |  |
| 15 | Common wallet not supported at this site. |  |  |
| 19 | Coupon Balance is Invalid. This error code is passed back when the parameter entered for couponamt has not met the criteria specified. |  |  |
| 21 | Integration Type is Invalid. The integrationtype parameter is only valid with value=2 and 6. |  |  |

Example

|  |
| --- |
| <?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>  <page>  <attribute>  <name>uid</name>  <value>1536</value>  </attribute>  <attribute>  <name>playerhandle</name>  <value>353535567317515965986999633642932172</value>  </attribute>  <attribute>  <name>clientAccount</name>  <value>1536</value>  </attribute>  <attribute>  <name>nickname</name>  <value>bigspender</value>  </attribute>  </page> |

|  |
| --- |
| <?xml version=”1.0” encoding=”ISO-8859-1” ?>  <page>  <attribute>  <name>playerhandle</name>  <value></value>  </attribute>  <attribute>  <name>uid</name>  <value></value>  </attribute>  </page> |

* + 1. GameLaunch

This API is used to get a lobby URL.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameSuite | mandatory | string | GameSuite specifies from which suite of games the game is to be loaded. This is always Flash. |
| playerHandle | mandatory | string | playerhandle in 4.5.1 Login() response  The playerHandle represents the logged in session of the player. Without a valid playerHandle, no games can be started. You can generate a playerHandle using the login method. |
| account | mandatory | string | player currency ISO code |
| gameId | optional | string | This gameId is a unique ID assigned to a given game. |
| gameName | optional | string | The gameName is the name that was assigned to the game. |
| gameType | optional | string | Some games have multiple variations. The gameType specifies the variation of a game to start. See the games list containing game IDs and game Types for this value. |
| firstName | optional | string | Guest will be used instead if it is not passed in. |
| lastName | optional | string | Guest will be used instead if it is not passed in. |
| lang | optional | string | gspMember Browser Language Code.  Lang specifies which language is to be used within the game. The table below specifies the language code to use. English is used when lang is not included.   |  |  | | --- | --- | | **Language** | **Code** | | Catalan | ca | | Chinese (Simplified) | zh-cn | | Chinese (Traditional) | zh-tw | | Czech | cz | | Danish | da | | Dutch | nl | | English | en | | French | fr | | German | de | | Greek | el | | Hebrew | he | | Hungarian | hu | | Italian | it | | Japanese | ja | | Korean | ko | | Norwegian | no | | Polish | pl | | Portuguese | pt | | Romanian | ro | | Russian | ru | | Slovak | sk | | Spanish | es | | Swedish | sv | | Turkish | tr | |
| theProfileId | optional | string | Use the Profile Id number found in CyberBoss or the database. The game will load with a specific profile. Otherwise the player will select a profile when the game loads. |
| denomination | optional | string | For games with multiple coin values, this will load the game with the specified denomination. |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameURL | mandatory | string | currency ISO code |

Example

|  |
| --- |
| ~?&gameSuite=Flash&playerHandle={playerHandle}&account={account} |

* + 1. Logout

Mount Point: ~/?method=**logout**&ph={**ph**}&account={**account}**

The API is used to log a gspMember out of TTGS.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| method | mandatory | string | logout |
| xmlresponse | mandatory | string | true  Ensures that the response is returned in XML format. Otherwise, the API will resort to using HTTP error codes which are less descriptive and harder to code against. Disabling this is allowed only for backward compatibility with legacy systems. |
| ph | mandatory | string | gspMember player handle |
| account | mandatory | string | parameter?  single wallet |

Example

|  |
| --- |
|  |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameURL | mandatory | string | game URL |

|  |  |  |  |
| --- | --- | --- | --- |
| **ERROR CODE** | **DESCRIPTION** | **REASON CODE** | **DESCRIPTION** |
| 1 | Client has logged out |  |  |
| 2 | No playerHandle passed in |  |  |
| 3 | Neither account nor attribute list present |  |  |
| 4 | Missing Required Fields |  |  |
| 5 | Player not found |  |  |

Example

|  |
| --- |
| <?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>  <page>  <attribute>  <name>USD;balance</name>  <value>59.67</value>  </attribute>  </page> |

* + 1. Exist

Mount Point: ~/?method=**exist**&uid={**uid}**

The API is used to check whether a gspMember exists in (or has ever logged into) TTG.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| method | mandatory | string | exist |
| xmlresponse | mandatory | string | true  Ensures that the response is returned in XML format. Otherwise, the API will resort to using HTTP error codes which are less descriptive and harder to code against. Disabling this is allowed only for backward compatibility with legacy systems. |
| uid | mandatory | string | gspMember id |

Example

|  |
| --- |
|  |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameURL | mandatory | string | game URL |

|  |  |  |  |
| --- | --- | --- | --- |
| **ERROR CODE** | **DESCRIPTION** | **REASON CODE** | **DESCRIPTION** |
| 4 | Missing Required Fields |  |  |
| 5 | Player not found |  |  |

Example

|  |
| --- |
| <?xml version=”1.0” encoding=”ISO-8859-1” ?>  <page>  <attribute>  <name>message</name>  <value>exists</value>  </attribute>  </page> |

* + 1. GetAccountNames

Mount Point: ~/?method=**getAccountNames**&ph={**ph}**

The API is used to retrieve a list of available accounts for a gspMember.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| method | mandatory | string | getAccountNames |
| xmlresponse | mandatory | string | true  Ensures that the response is returned in XML format. Otherwise, the API will resort to using HTTP error codes which are less descriptive and harder to code against. Disabling this is allowed only for backward compatibility with legacy systems. |
| ph | mandatory | string | gspMember player handle |

Example

|  |
| --- |
|  |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameURL | mandatory | string | game URL |

|  |  |  |  |
| --- | --- | --- | --- |
| **ERROR CODE** | **DESCRIPTION** | **REASON CODE** | **DESCRIPTION** |
| 2 | No Player Handle Present |  |  |
| 4 | Missing Required Fields |  |  |
| 10 | Unable to Get Account Names |  |  |

Example

|  |
| --- |
| <?xml version=”1.0” encoding=”ISO-8859-1” ?>  <page>  <attribute>  <name>playerhandle</name>  <value></value>  </attribute>  <attribute>  <name>accountnames</name>  <value></value>  </attribute>  </page> |

* + 1. UpdatePlayer

Mount Point: ~/?method=**getAccountNames**&ph={**ph}**

The API is used to update a gspMember’s information.

The system does not compare the old value with new value, but simply uses the new value. Some player information parameters cannot be changed.

These include: userName, nickName, balance, and account.

They will be ignored if passed in, not considered errors.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| method | mandatory | string | updatePlayer |
| xmlresponse | mandatory | string | true  Ensures that the response is returned in XML format. Otherwise, the API will resort to using HTTP error codes which are less descriptive and harder to code against. Disabling this is allowed only for backward compatibility with legacy systems. |
| uid | mandatory | string | gspMember id |
| active | optional | string | false: disable the player  Any other value will enable the player. |
| currency | optional | string | This can be updated, but the player's account cannot. In the event that the currency does not match the player's account, the currency will overwrite the account. |
| country | optional | string | This field can be updated. Anything entered here will replace whatever country code was entered previously.  format: ISO 3166 alpha-2  Use xx as the country code if the player is FunMoney only. |
| fn | optional | string | gspMember first name  This field can be updated. First name, (max. 50 characters), defaults to the unique identifier if not specified. Anything entered here will become the player’s first name, replacing whatever was entered before. |
| ln | optional | string | gspMember last name.  This field can be updated. Last name, (max. 50 characters), defaults to the unique identifier if not specified. Anything entered here will become the player's last name, replacing whatever was entered before. |

Example

|  |
| --- |
|  |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
|  |  |  |  |

Example

|  |
| --- |
| <page>  <attribute>  <name/>  <value/>  </attribute>  <attribute>  <name/>  <value/>  </attribute>  ...  </page> |

* + 1. Account Balance Detail (Unused)

Mount Point: https://<privateServer>/sdk/servlet/com.cwh.accounting.PlayerInfoServlet

When a user logs out of the casino currently, they receive the option to transfer funds out of the casino. Players will be informed if they have to leave a certain amount in the casino to keep their bonus in play. The external system will make a call to the Top Trend system and retrieve the exact amount necessary. This web service retrieves the player’s casino balance information.

* + 1. GetPlayerInformation

Mount Point: ~/?playerHandle={**playerHandle**}&method=getallinfo

The API is used to update a gspMember’s information.

This API is used to get an information of gspMember.

If the method is not **getallinfo**, then then only critical or required information will be returned.

If **method=getallinfo** is used, then general player attributes will also be returned

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| method | optional | string | getallinfo |
| playerHandle | mandatory | string | gspMember player handle |

Example

|  |
| --- |
| http://casinoHost/sdk/servlet/com.chartwelltechnology.icd.ClientInfoServlet?playerHandle=<playerHandle>&method=getallinfo |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
|  |  |  |  |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?>  <page>  <attribute>  <name>playerHandle</name>  <value></name>  </attribute>  <attribute>  <name>clientAccount</name>  <value></name>  </attribute>  <attribute>  <name>regAffId</name>  <value></name>  </attribute>  <attribute>  <name>firstName</name>  <value></name>  </attribute>  ...  </page> |

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <page>  <error>playerHandleMissed</error>  </page> |

Invalid method (the only method supported is **getallinfo**, so anything else is considered invalid):

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <page>  <error>invalidMethod</error>  </page> |

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <page>  <error>playerHandleNotFound</error>  </page> |

* + 1. Coded Bonus (Unused)

The Gaming System supports the operator to pass in a predefined bonus code thus allowing the player to receive the coded bonus. The coded bonus plan is set up in CyberBoss. See the CyberBoss manual for directions on setting up a coded bonus plan.

* + 1. GetBalance (From TTG to IMS)

Mount Point: ~/GetBalance

The API is used to retrieve a player balance in IMS.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| acctid | mandatory | string | TTG member account number  = client account number in Login() response |
| playerHandle | mandatory | string(36) | gspMember player handle |
| gameid | optional | int | TTG game id |
| cur | mandatory | string | currency ISO code  Currency (ISO 3 char code) that the player is using to play the game and must be the currency the player used during registration/login. |

Example

|  |
| --- |
| <cw  type=”**getBalanceReq**”  acctid=”abc123”  cur=”EUR”  playerhandle=”[157157100004873621461557667327238820]”” /> |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| cur | mandatory | string | currency ISO code |
| amt | mandatory | decimal | Amount of the players balance in the external wallet. Reponse for amount should be in decimal format e.g. 1000.00 |
| err | mandatory | string | Error code is returned with each response |

Example

|  |
| --- |
| <cw type=”getBalanceResp” cur=”EUR” amt=”100.00” err=”0” /> |

|  |
| --- |
| <cw type=”getBalanceResp” err=”1000” /> |

|  |  |
| --- | --- |
| **ERROR CODE** | **DESCRIPTION** |
| 0 | no error |
| 1000 | account does not exist |
| 1001 | invalid currency |
| 9999 | unknown error |

* + 1. FundTransfer (From TTG to IMS)

Mount Point: ~/FundTransfer

The API is used to credit transfer or to debit transfer by TTG.

The TTG makes this request of IMS to modify the balance of the player, either to withdraw funds for a bet, or to deposit funds for a win.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| acctid | mandatory | string | TTG member account number  = client account number in Login() response  = same as uid? |
| cur | mandatory | string | currency ISO code  Currency (ISO 3 char code) that the player is using to play the game and must be the currency the player used during registration/login. |
| amt | mandatory | int | Amount of the transaction to adjust the player’s balance by.  The sign will determine if it is an increase or decrease  -: Debit(Withdraw)  +: Credit(Deposit)  Maximum of 2 decimal places. |
| txnid | mandatory | long | transaction id  This ID must be stored in the IMS database as it will be used to validate if this is a new request or a retry. |
| canceltxnid | optional | long | Specifies the txnId that this request is referenced to. This attribute will only be sent if a cancel request is made (txnsubtypeid: Bet, and amt: positive). |
| gameid | optional | int | TTG game id |
| txnsubtypeid | optional | int | TTG transaction sub type id  400 – Bet (if Bet is positive amount, it is considered a cancelled bet)  410 – Win  150 – Bonus  151 – Bonus Adjustment  420 – Game Adjustment  450 – External Game Bet  460 – External Game Win  600 – PJ Win |
| handid | optional | int | TTG game hand id |
| playerHandle | optional | string(36) | TTGS playerhandle (optional)  for fundtransfer requests where no playerhandle is known or given, the value may be 0, N/A or not set. |

Example

|  |
| --- |
| <cw type=”fundTransferReq” acctid=”[abc123]” cur=”[EUR]” amt=”[100.00]” txnid=”[1000]”  gameid=”[5]” txnsubtypeid=”[400]” handid=”[1]”  playerhandle=”[157157100004873621461557667327238820]”/> |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| cur | mandatory | string | currency ISO code |
| amt | mandatory | decimal | Amount of the players balance in the IMS.  Reponse for amount should be in decimal format e.g. 1000.00 |
| refid | optional | string | IMS transaction id |
| err | mandatory | string | Error code is returned with each response |

Example

|  |
| --- |
| cw type=”fundTransferResp” cur=”EUR” amt=”100.00” err=”0” |

|  |
| --- |
| cw type=”fundTransferResp” err=”1000” |

|  |  |
| --- | --- |
| **ERROR CODE** | **DESCRIPTION** |
| 0 | no error |
| 1000 | account does not exist |
| 1001 | invalid currency |
| 1002 | invalid amount |
| 1004 | locked account |
| 9999 | unknown error |

* + 1. FundTransfer – Multiple Transactions (From TTG to IMS)

Mount Point: ~/FundTransfer/MultipleTransactions

The API is used to credit transfer or to debit transfer by TTG.

The TTG makes this request of IMS to modify the balance of the player, either to withdraw funds for a bet, or to deposit funds for a win.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| acctid | mandatory | string | TTG member account number  = client account number in Login() response |
| cur | mandatory | string | currency ISO code  Currency (ISO 3 char code) that the player is using to play the game and must be the currency the player used during registration/login. |
| amt | mandatory | int | Amount of the transaction to adjust the player’s balance by.  The sign will determine if it is an increase or decrease  -: Debit(Withdraw)  +: Credit(Deposit)  Maximum of 2 decimal places. |
| txnid | mandatory | long | transaction id  This ID must be stored in the IMS database as it will be used to validate if this is a new request or a retry. |
| canceltxnid | optional | long | Specifies the txnId that this request is referenced to. This attribute will only be sent if a cancel request is made (txnsubtypeid: Bet, and amt: positive). |
| gameid | optional | int | TTG game id |
| txnsubtypeid | optional | int | TTG transaction sub type id  400 – Bet (if Bet is positive amount, it is considered a cancelled bet)  410 – Win  150 – Bonus  151 – Bonus Adjustment  420 – Game Adjustment  450 – External Game Bet  460 – External Game Win  600 – PJ Win |
| handid | optional | int | TTG game hand id |
| playerHandle | optional | string(36) | TTGS playerhandle (optional)  for fundtransfer requests where no playerhandle is known or given, the value may be 0, N/A or not set. |
| transactions | optional | int | the number of transactions that the request included |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?>  <cw type=”fundTransferReq” acctid=”abc123” transactions=”2”>  <detail amt="0.15" cur=”EUR” txnid="2" gameid="406" txnsubtypeid="410" handid="1" playerhandle=”157157100004873621461557667327238820” />  <detail amt="-0.15" cur=”EUR” txnid="3" gameid="402" txnsubtypeid="400" handid="2" playerhandle=”157157100004873621461557667327238820” />  </cw> |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| cur | mandatory | string | currency ISO code |
| amt | mandatory | decimal | Amount of the players balance in the IMS.  Reponse for amount should be in decimal format e.g. 1000.00 |
| refid | optional | string | IMS transaction id |
| err | mandatory | string | Error code is returned with each response |

Example

|  |
| --- |
| cw type=”fundTransferResp” cur=”EUR” amt=”100.00” err=”0” |

|  |
| --- |
| cw type=”fundTransferResp” err=”1000” |

|  |  |
| --- | --- |
| **ERROR CODE** | **DESCRIPTION** |
| 0 | no error |
| 1000 | account does not exist |
| 1001 | invalid currency |
| 1002 | invalid amount |
| 1004 | locked account |
| 9999 | unknown error |

* + 1. FundTransfer – CombinedTransactions (From TTG to IMS)

Mount Point: ~/FundTransfer

The API is used to credit transfer or to debit transfer by TTG.

The TTG makes this request of IMS to modify the balance of the player, either to withdraw funds for a bet, or to deposit funds for a win.

For the combine transaction request:

amt is the total amount for all the transactions involved.

If the total amount = 0, no fund transfer request will be sent to the integrated site (by default), but can be configured to do so.

If the total amount < 0, the fund transfer request will behave like a bet request (i.e. handid, txnid, txnsubtypeid, gameid is coming from the bet transaction).

If the total amount > 0, the fund transfer request will behave like a winning request (i.e. handid, txnid, txnsubtypeid, gameid is coming from the winning transaction).

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| acctid | mandatory | string | TTG member account number  = client account number in Login() response |
| cur | mandatory | string | currency ISO code  Currency (ISO 3 char code) that the player is using to play the game and must be the currency the player used during registration/login. |
| amt | mandatory | int | Amount of the transaction to adjust the player’s balance by.  The sign will determine if it is an increase or decrease  -: Debit(Withdraw)  +: Credit(Deposit)  Maximum of 2 decimal places. |
| txnid | mandatory | long | transaction id  This ID must be stored in the IMS database as it will be used to validate if this is a new request or a retry. |
| canceltxnid | optional | long | Specifies the txnId that this request is referenced to. This attribute will only be sent if a cancel request is made (txnsubtypeid: Bet, and amt: positive). |
| gameid | optional | int | TTG game id |
| txnsubtypeid | optional | int | TTG transaction sub type id  400 – Bet (if Bet is positive amount, it is considered a cancelled bet)  410 – Win  150 – Bonus  151 – Bonus Adjustment  420 – Game Adjustment  450 – External Game Bet  460 – External Game Win  600 – PJ Win |
| handid | optional | int | TTG game hand id |
| playerHandle | optional | string(36) | TTGS playerhandle (optional)  for fundtransfer requests where no playerhandle is known or given, the value may be 0, N/A or not set. |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <cw type=”fundTransferReq” acctid=”abc123” cur=”EUR” amt=”0.07” txnid=”2” gameid=”406” txnsubtypeid=”410” handid=”1” playerhandle=”157157100004873621461557667327238820”/> |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| cur | mandatory | string | currency ISO code |
| amt | mandatory | decimal | Amount of the players balance in the IMS.  Reponse for amount should be in decimal format e.g. 1000.00 |
| refid | optional | string | IMS transaction id |
| err | mandatory | string | Error code is returned with each response |

Example

|  |
| --- |
| cw type=”fundTransferResp” cur=”EUR” amt=”100.00” err=”0” |

|  |
| --- |
| cw type=”fundTransferResp” err=”1000” |

|  |  |
| --- | --- |
| **ERROR CODE** | **DESCRIPTION** |
| 0 | no error |
| 1000 | account does not exist |
| 1001 | invalid currency |
| 1002 | invalid amount |
| 1004 | locked account |
| 9999 | unknown error |

* + 1. FundTransfer – Net Zero Transactions (From TTG to IMS)

Mount Point: ~/FundTransfer

The API is used to credit transfer or to debit transfer by TTG.

The TTG makes this request of IMS to modify the balance of the player, either to withdraw funds for a bet, or to deposit funds for a win.

Normally Common Wallet service will not send request to common wallet site when there is no balance change. However, if requested, a Fund transfer request of amount “0” can be sent when a game hand is a loss (or net zero for any reason). This can be used to simply indicate the end of the hand.

By definition, a losing hand: is considered as a winning transaction with amount 0

Like other fund transfer request, TTGS will expect response in the same xml format as a Bet or Win request. Based on the response, TTGS will handle in the following ways:

a. TTGS receives a successful response – the game continues.

b. TTGS receives an error response (e.g. with error code other than system error) - the game continues; the error will be logged for future troubleshooting.

c. TTGS receives system error or communication timeout error – the game continues; TTGS will retry later.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| acctid | mandatory | string | TTG member account number  = client account number in Login() response |
| cur | mandatory | string | currency ISO code  Currency (ISO 3 char code) that the player is using to play the game and must be the currency the player used during registration/login. |
| amt | mandatory | int | Amount of the transaction to adjust the player’s balance by.  The sign will determine if it is an increase or decrease  -: Debit(Withdraw)  +: Credit(Deposit)  Maximum of 2 decimal places. |
| txnid | mandatory | long | transaction id  This ID must be stored in the IMS database as it will be used to validate if this is a new request or a retry. |
| canceltxnid | optional | long | Specifies the txnId that this request is referenced to. This attribute will only be sent if a cancel request is made (txnsubtypeid: Bet, and amt: positive). |
| gameid | optional | int | TTG game id |
| txnsubtypeid | optional | int | TTG transaction sub type id  400 – Bet (if Bet is positive amount, it is considered a cancelled bet)  410 – Win  150 – Bonus  151 – Bonus Adjustment  420 – Game Adjustment  450 – External Game Bet  460 – External Game Win  600 – PJ Win |
| handid | optional | int | TTG game hand id |
| playerHandle | optional | string(36) | TTGS playerhandle (optional)  for fundtransfer requests where no playerhandle is known or given, the value may be 0, N/A or not set. |

Example

|  |
| --- |
| <cw type="fundTransferReq" acctid="cwTest21" cur="USD" amt="0" txnid="25" gameid="5" txnsubtypeid="410" /> |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| cur | mandatory | string | currency ISO code |
| amt | mandatory | decimal | Amount of the players balance in the IMS.  Reponse for amount should be in decimal format e.g. 1000.00 |
| refid | optional | string | IMS transaction id |
| err | mandatory | string | Error code is returned with each response |

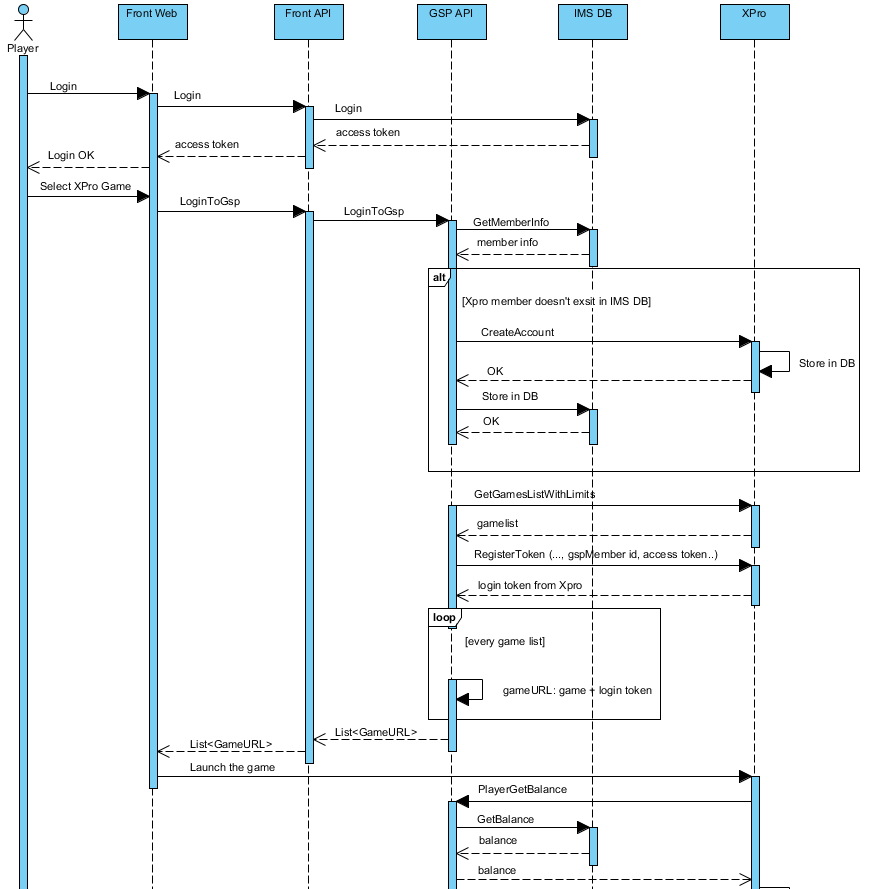
Example

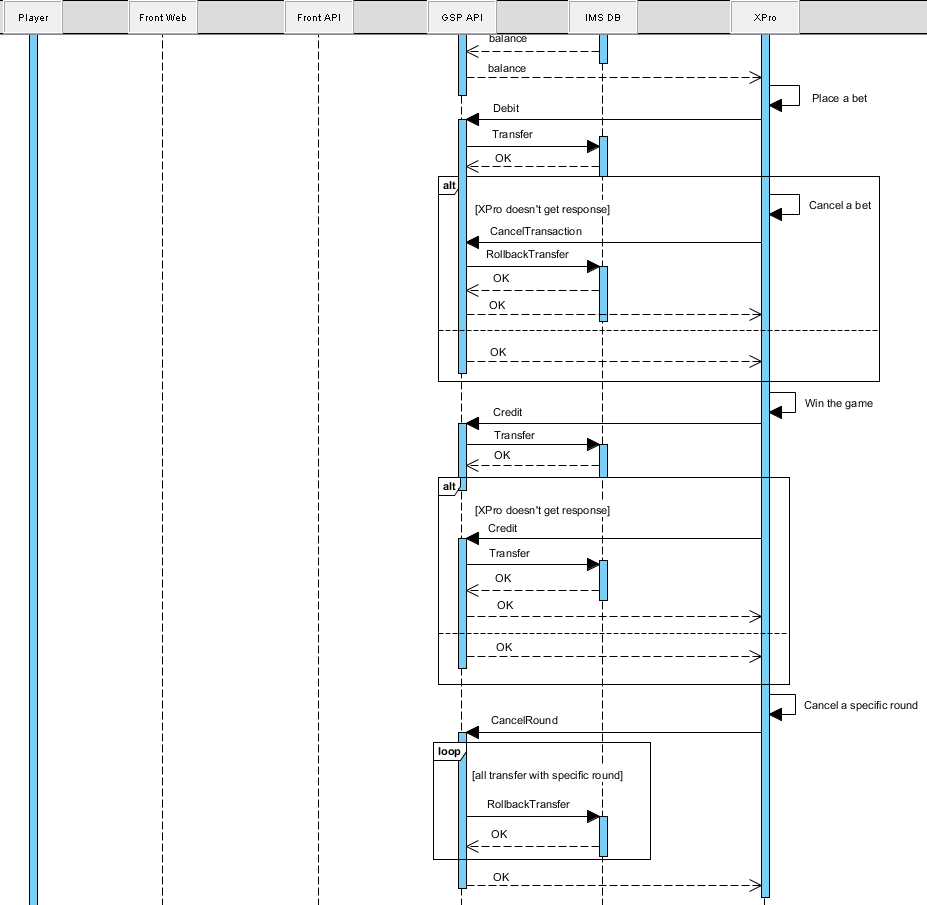
|  |
| --- |
| cw type=”fundTransferResp” cur=”EUR” amt=”100.00” err=”0” |

|  |
| --- |
| cw type=”fundTransferResp” err=”1000” |

|  |  |
| --- | --- |
| **ERROR CODE** | **DESCRIPTION** |
| 0 | no error |
| 1000 | account does not exist |
| 1001 | invalid currency |
| 1002 | invalid amount |
| 1004 | locked account |
| 9999 | unknown error |

* 1. XPro Gaming (XPRO)





|  |  |  |
| --- | --- | --- |
| **Reference Document** | **Version** | **File Name** |
| **Integration API documentation.** | **V 3.4** | **[사용중]XProGamingAPIDocV3.4.pdf** |
| **XProGaming API Design – Seamless Wallet** | **v.1.5.4** | **XProGaming API - Seamless WalletV1.5.4.pdf** |

* + 1. CreateAccount

Mount Point: ~/createAccount

The API is used to create new member account on XPRO.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| operatorId | mandatory | int | gspGameCurrencyId |
| username | mandatory | string | gspMember id  Has to be unique for operator. |
| accessPassword | mandatory | string | validation key  ex) privateKey: opToGspCurrency password  urlParam = “operatorId=” + gspGameCurrencyId  urlParam += “&username=” gspMemberId  accessPassword = *MD5*(privateKey + urlParam) |
| userPassword | optional | string | gspMember’s password(MD5) |
| email | optional | string | gspMember’s email  Has to be unique for operator. |
| firstName | optional | string | gspMember’s first name |
| lastName | optional | string | gspMember’s last name |
| accountID | optional | int | External system ID of the account.  Has to be unique for operator. |
| isExteranlWallet | optional | int | Is it member using seamless wallet? |
| Nickname | optional | string | Nickname to be shown in the game.  Otherwise username will be used. |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errorCode | mandatory | int | error code |
| description | mandatory | string | If necessary, a short description of the error. |

Example

|  |
| --- |
| <response xmlns=”apiResultData” xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance” xmlns:xsd=”http://www.w3.org/2001/XMLSchema”>  <errorCode>0</errorCode>  <description />  </response> |

|  |  |  |
| --- | --- | --- |
| **ERROR CODE** | **ERROR Name** | **DESCRIPTION** |
| 0 | NoError | Successful call |
| 3 | PlayerAlreadyExists | Player record already exists |
| 4 | NicknameAlreadyExists | Player with the same nickname exists |
| 5 | RequiredFieldMissing | Obligatory field missing |
| 6 | AuthenticationFailed | Sent access password does not match the server |
| 16 | EmailAlreadyExists | Player with the same email already exists |
| 100 | UnspecifiedError | Unknown error |

* + 1. ChangeAccount

Mount Point: ~/ChangeAccount

The API is used to change properties of existing member in XPro.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| operatorId | mandatory | int | gspGameCurrencyId |
| username | mandatory | string | gspMember id  Has to be unique for operator. |
| accessPassword | mandatory | string | validation key  ex) privateKey: opToGspCurrency password  urlParam = “operatorId=” + gspGameCurrencyId  urlParam += “&username=” gspMemberId  accessPassword = *MD5*(privateKey + urlParam) |
| newPassword | optional | string | New gspMember password, MD5 hashed. |
| newNickname | optional | string | New nickname of the gspMember.  Will be shown in the game. |
| newStatus | optional | int | new status of the gspMember  1: Active  2: Suspended  3: Canceled |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errorCode | mandatory | int | error code |
| description | mandatory | string | If necessary, a short description of the error. |

Example

|  |
| --- |
| <response xmlns=”apiResultData” xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance” xmlns:xsd=”http://www.w3.org/2001/XMLSchema”>  <errorCode>0</errorCode>  <description />  </response> |

|  |  |  |
| --- | --- | --- |
| **ERROR CODE** | **ERROR Name** | **DESCRIPTION** |
| 0 | NoError | Successful call |
| 3 | PlayerAlreadyExists | Player record already exists |
| 4 | NicknameAlreadyExists | Player with same nickname exists |
| 5 | RequiredFieldMissing | Obligatory field missing |
| 6 | AuthenticationFailed | Sent access password does not match the server |
| 16 | EmailAlreadyExists | Player with the same email already exists |
| 100 | UnspecifiedError | Unknown error |

* + 1. KickPlayerFromGames

Mount Point: ~/createAccount

The API is used to create new member account on XPRO.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| operatorId | mandatory | int | gspGameCurrencyId |
| username | mandatory | string | gspMember id  Has to be unique for operator. |
| accessPassword | mandatory | string | validation key  ex) privateKey: opToGspCurrency password  urlParam = “operatorId=” + gspGameCurrencyId  urlParam += “&username=” gspMemberId  accessPassword = *MD5*(privateKey + urlParam) |
| bulkOperation | optional | int | indicates whether the list of usernames is compressed. Default is 0(not compressed). If the username param contains more than 1 user, bulkOperation param must have value 1 otherwise only the first user will be affected  0: not compressed(default)  1: compressed |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errorCode | mandatory | int | error code |
| description | mandatory | string | If necessary, a short description of the error. |

Example

|  |
| --- |
| <response xmlns=”apiResultData” xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance” xmlns:xsd=”http://www.w3.org/2001/XMLSchema”>  <errorCode>0</errorCode>  <description />  </response> |

|  |  |  |
| --- | --- | --- |
| **ERROR CODE** | **ERROR Name** | **DESCRIPTION** |
| 0 | NoError | Successful call |
| 2 | PlayerNotFound | No player with the sent username found |
| 5 | RequiredFieldMissing | Obligatory field missing |
| 6 | AuthenticationFailed | Sent access password does not match the server |
| 24 | PlayerKickFailed | Players failed to be kicked from games. (might be in compressed mode depending on bulkOperation param value) |
| 27 | FailedToDecompressData | Fails to decompress username list. |
| 100 | UnspecifiedError | Unknown error |

* + 1. ChangeAccountStatus

Mount Point: ~/ChangeAccountStatus

The API is used to change a gspMember’s status in XPro.

This API can be used to ban/unban gspMembers on the XPro.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| operatorId | mandatory | int | gspGameCurrencyId |
| username | mandatory | string | gspMember id  Has to be unique for operator. |
| accessPassword | mandatory | string | validation key  ex) privateKey: opToGspCurrency password  urlParam = “operatorId=” + gspGameCurrencyId  urlParam += “&username=” gspMemberId  accessPassword = *MD5*(privateKey + urlParam) |
| newStatus | mandatory | int | new status of the gspMember  1: Active  2: Suspended  3: Canceled |
| bulkOperation | optional | int | indicates whether the list of usernames is compressed. Default is 0(not compressed). If the username param contains more than 1 user, bulkOperation param must have value 1 otherwise only the first user will be affected  0: not compressed(default)  1: compressed |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errorCode | mandatory | int | error code |
| description | mandatory | string | If necessary, a short description of the error. |

Example

|  |
| --- |
| <response xmlns=”apiResultData” xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance” xmlns:xsd=”http://www.w3.org/2001/XMLSchema”>  <errorCode>0</errorCode>  <description />  </response> |

|  |  |  |
| --- | --- | --- |
| **ERROR CODE** | **ERROR Name** | **DESCRIPTION** |
| 0 | NoError | Successful call |
| 2 | PlayerNotFound | No player with the sent username found |
| 5 | RequiredFieldMissing | Obligatory field missing |
| 6 | AuthenticationFailed | Sent access password does not match the server |
| 15 | ErrorUpdatingPlayer | Players update failed. (might be in compressed mode depending on bulkOperation param value) |
| 27 | FailedToDecompressData | Fails to decompress username list. |
| 100 | UnspecifiedError | Unknown error |

* + 1. ChangeAccountPassword

Mount Point: ~/ChangeAccountPassword

The API is used to change a gspMember’s password in XPro.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| operatorId | mandatory | int | gspGameCurrencyId |
| username | mandatory | string | gspMember id  Has to be unique for operator. |
| accessPassword | mandatory | string | validation key  ex) privateKey: opToGspCurrency password  urlParam = “operatorId=” + gspGameCurrencyId  urlParam += “&username=” gspMemberId  accessPassword = *MD5*(privateKey + urlParam) |
| newPassword | mandatory | string | updated password for the gspMember(MD5) |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errorCode | mandatory | int | error code |
| description | mandatory | string | If necessary, a short description of the error. |

Example

|  |
| --- |
| <response xmlns=”apiResultData” xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance” xmlns:xsd=”http://www.w3.org/2001/XMLSchema”>  <errorCode>0</errorCode>  <description />  </response> |

|  |  |  |
| --- | --- | --- |
| **ERROR CODE** | **ERROR Name** | **DESCRIPTION** |
| 0 | NoError | Successful call |
| 2 | PlayerNotFound | No player with the sent username found |
| 5 | RequiredFieldMissing | Obligatory field missing |
| 6 | AuthenticationFailed | Sent access password does not match the server |
| 15 | ErrorUpdatingPlayer | Players update failed. (might be in compressed mode depending on bulkOperation param value) |
| 100 | UnspecifiedError | Unknown error |

* + 1. GetGamesListWithLimits

Mount Point: ~/getGamesListWithLimits

The API is used to get the list of games currently defined.

This list can include both Online and Offline games according to the parameters supplied.

The List includes three types of data:

• Game limits definitions

• The game information such as game type, game name, opening hours etc.

• The game opening URL

The list can be provided in two modes – “normal mode” and “viewer mode”.

The difference between the two modes is that in viewer mode the game information which is specific to the player (such as limits) has a default value. Also, the game opening URL is slightly different.

The game data should be displayed to the players in any form dictated by the design and UI requirements of the hosting web site. The players should be allowed to choose a game and open it. The method of handling and opening the game with the game opening URL is described in Appendix C.

Regarding the option of opening games with different limits, when registering a token with the limit set ID that applies to the game, a player can play the game with the limits that are defined in the selected limit set.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| operatorId | mandatory | int | gspGameCurrencyId |
| username | mandatory | string | gspMember id  Has to be unique for operator.  Can be empty to indicate the lobby should be displayed in “Viewer Mode” |
| accessPassword | mandatory | string | validation key  ex) privateKey: opToGspCurrency password  urlParam = “operatorId=” + gspGameCurrencyId  urlParam += “& &gameProvider=All”  accessPassword = *MD5*(privateKey + urlParam) |
| gameType | mandatory | int | the type of the game to get  0: hard-code |
| onlineOnly | mandatory | boolean | show only games which are online now  0: hard-code |
| includeInOutLimits | optional | int | whether to include inside and outside total limits for roulette, and member bet minimum and maximum for baccarat and sic-bo |
| includeBaccaratBigRoadMap | optional | Boolean | whether to include big road map for baccarat |
| includeDragonTigerBigRoadMap | optional | Boolean | whether to include big road map for Dr Tg |
| includeRouletteLastResults | optional | Boolean | whether to include last roulette results |
| includerouletteStatistics | optional | Boolean | whether to include roulette statistics |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errorCode | mandatory | int | error code |
| description | mandatory | string | If necessary, a short description of the error. |
| gameList | mandatory | List<Game> | container tag for all of games returned.  Empty tag when no games are returned |

Game

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| limitSetList | mandatory | LimitSetList |  |
| gameID | mandatory | int | the unique id for this game |
| gameType | mandatory | int | the type of the game as defined by the game types list |
| gameName | mandatory | string | the name of the game |
| dealerName | mandatory | string | the name of the dealer (empty for closed games) |
| dealerImageUrl | mandatory | string | dealer image link to use to |
| isOpen | mandatory | Boolean | whether or not the game is online(open) or not.  member should not be allowed to log into closed games  it will fail |
| connectionURL | mandatory | string | The Min/Max limit for the game  For Blackjack game – Per seat  For Baccarat and Sic-Bo games – Total on table  For Roulette – Straight up bet |
| minBet | mandatory | decimal |
| maxBet | mandatory | decimal |
| openHour | mandatory | time? | show the usual Opening/Closing hour for this game  GMT timezone |
| closeHour | mandatory | time? | show the usual Opening/Closing hour for this game  GMT timezone |
| bigRoadMap | mandatory | BigRoadMap |  |

LimitSetList

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| limitSet | mandatory | LimitSet |  |

LimitSet

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| limitSetID | mandatory | string |  |
| minBet | mandatory | string |  |
| maxBet | mandatory | string |  |
| minPlayerBet | mandatory | string |  |
| maxplayerBet | mandatory | string |  |

BigRoadMap

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| item | mandatory | GamePlayItem |  |

GamePlayItem

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| x | mandatory | string |  |
| y | mandatory | string |  |
| result | mandatory | string |  |
| score | mandatory | string |  |
| isBankerPair | mandatory | string |  |
| isPlayerPair | mandatory | string |  |

Example

|  |
| --- |
| <?xml version=”1.0” encoding=”utf-8”?>  <response xmlns=”apiGamesLimitsListData” xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance” xmlns:xsd=”http://www.w3.org/2001/XMLSchema”>  <gamesList>  <game>  <limitSetList>  <limitSet>  <limitSetID>2</limitSetID>  <minBet>10</minBet>  <maxBet>100</maxBet>  <minInsideBet>4</minInsideBet>  <maxInsideBet>402</maxInsideBet>  <minOutsideBet>4</minOutsideBet>  <maxOutsideBet>403</maxOutsideBet>  </limitSet>  </limitSetList>  <rouletteLastResults>  <item>23</item>  <item>15</item>  <item>10</item>  <item>6</item>  <item>14</item>  <item>3</item>  <item>32</item>  </rouletteLastResults>  <roulleteResultStatistics>  <roundNumber>300</roundNumber>  <pastResults>  <item>22</item>  <item>9</item>  <item>0</item>  <item>2</item>  <item>2</item>  <item>17</item>  <item>3</item>  <item>14</item>  <item>6</item>  <item>23</item>  <item>7</item>  <item>0</item>  </pastResults>  <hotNumbers>  <item>24</item>  <item>3</item>  <item>35</item>  <item>6</item>  <item>14</item>  <item>9</item>  </hotNumbers>  <hotNumbersCount>  <item>15</item>  <item>12</item>  <item>12</item>  <item>11</item>  <item>11</item>  <item>11</item>  </hotNumbersCount>  <coldNumbers>  <item>17</item>  <item>4</item>  <item>13</item>  <item>25</item>  <item>6</item>  <item>8</item>  </coldNumbers>  <coldNumbersCount>  <item>149</item>  <item>106</item>  <item>88</item>  <item>81</item>  <item>72</item>  <item>71</item>  </coldNumbersCount>  <blackPercent>50.33</blackPercent>  <redPercent>46.67</redPercent>  <oddPercent>48.33</oddPercent>  <evenPercent>48.67</evenPercent>  <zeroPercent>3.00</zeroPercent>  </roulleteResultStatistics>  <gameID>4</gameID>  <gameType>1</gameType>  <gameName>roulette1</gameName>  <dealerName>Dealer1</dealerName>  <dealerImageUrl> [http://lcppflash.xprogaming.com/dealer/1.jpg</dealerImageUrl](http://lcppflash.xprogaming.com/dealer/1.jpg%3c/dealerImageUrl)>  <isOpen>0</isOpen>  <connectionUrl>SEE\_APPENDIX\_C</connectionUrl>  <winParams>SEE\_APPENDIX\_C</winParams>  <openHour>00:00</openHour>  <closeHour>00:00</closeHour>  </game>  <game>  <limitSetList>  <limitSet>  <limitSetID>6</limitSetID>  <minBet>10</minBet>  <maxBet>50</maxBet>  <minPlayerBet>1</minPlayerBet>  <maxPlayerBet>50</maxPlayerBet>  </limitSet> </limitSetList>  <gameID>5</gameID>  <gameType>4</gameType>  <gameName>baccarat1</gameName>  <dealerName>Dealer2</dealerName>  <dealerImageUrl> [http://lcppflash.xprogaming.com/dealer/2.jpg</dealerImageUrl](http://lcppflash.xprogaming.com/dealer/2.jpg%3c/dealerImageUrl)>  <isOpen>0</isOpen>  <connectionUrl>SEE\_APPENDIX\_C</connectionUrl>  <winParams>SEE\_APPENDIX\_C</winParams>  <openHour>11:00</openHour>  <closeHour>22:00</closeHour>  <bigRoadMap>  <item>  <X>1</X>  <Y>1</Y>  <Result>2</Result>  <Score>6</Score>  <isBankerPair>1</isBankerPair>  <isPlayerPair>0</isPlayerPair>  </item>  <item> <X>1</X> <Y>2</Y>  <Result>2</Result>  <Score>3</Score> <isBankerPair>0</isBankerPair>  <isPlayerPair>0</isPlayerPair> </item>  <item> <X>2</X> <Y>1</Y>  <Result>1</Result>  <Score>7</Score> <isBankerPair>0</isBankerPair>  <isPlayerPair>1</isPlayerPair> </item>  <item> <X>2</X> <Y>2</Y>  <Result>3</Result>  <Score>2</Score> <isBankerPair>0</isBankerPair>  <isPlayerPair>0</isPlayerPair> </item>  </bigRoadMap>  </game>  <game>  <limitSetList>  <limitSet>  <limitSetID>56</limitSetID>  <minBet>50.00</minBet>  <maxBet>1000.00</maxBet>  </limitSet>  </limitSetList>  <DTBigRoadMap>  <item>  <X>0</X>  <Y>0</Y>  <Result>1</Result>  <Score>13</Score>  </item>  <item>  <X>1</X>  <Y>0</Y>  <Result>2</Result>  <Score>7</Score>  </item>  <item>  <X>1</X>  <Y>1</Y>  <Result>2</Result>  <Score>4</Score>  </item>  </DTBigRoadMap>  <gameID>40</gameID>  <gameType>12</gameType>  <gameName>Dragon Tiger Test 2</gameName>  <dealerName>Dealer</dealerName>  <dealerImageUrl>[http://lcppflash.xprogaming.com/dealer/3.jpg</dealerImageUrl](http://lcppflash.xprogaming.com/dealer/3.jpg%3c/dealerImageUrl)>  <isOpen>1</isOpen>  <connectionUrl>SEE\_APPENDIX\_C</connectionUrl>  <winParams>SEE\_APPENDIX\_C</winParams>  <openHour>00:00</openHour>  <closeHour>23:59</closeHour>  <PlayersNumberxsi:nil=”true”/>  <PlayersNumberInGamexsi:nil=”true”/>  </game>  <game>  <limitSetList>  <limitSet>  <limitSetID>6</limitSetID>  <minBet>10</minBet>  <maxBet>50</maxBet>  </limitSet> <limitSet>  <limitSetID>10</limitSetID>  <minBet>20</minBet>  <maxBet>200</maxBet>  <minBetBehind>10</minBetBehind>  <maxBetBehind>100</maxBetBehind>  </limitSet>  </limitSetList>  <gameID>6</gameID>  <gameType>2</gameType>  <gameName>blackjack1</gameName>  <dealerName>Dealer3</dealerName>  <dealerImageUrl> [http://lcppflash.xprogaming.com/dealer/3.jpg</dealerImageUrl](http://lcppflash.xprogaming.com/dealer/3.jpg%3c/dealerImageUrl)>  <isOpen>0</isOpen>  <connectionUrl>SEE\_APPENDIX\_C</connectionUrl>  <winParams>SEE\_APPENDIX\_C</winParams>  <openHour>11:00</openHour>  <closeHour>22:00</closeHour> <PlayersNumber>7</PlayersNumber> <PlayersNumberInGame>0</PlayersNumberInGame>  </game>  <game>  <limitSetList>  <limitSet>  <limitSetID>60</limitSetID>  <minBet>5.00</minBet>  <maxBet>1000.00</maxBet>  </limitSet>  </limitSetList>  <gameID>42</gameID>  <gameType>16</gameType>  <gameName>Live Sic Bo 1 </gameName>  <dealerName>Dealer</dealerName>  <dealerImageUrl>url</dealerImageUrl>  <isOpen>1</isOpen>  <connectionUrl> SEE\_APPENDIX\_C </connectionUrl>  <winParams> SEE\_APPENDIX\_C </winParams>  <openHour>09:00</openHour>  <closeHour>04:00</closeHour>  <PlayersNumber xsi:nil=”true”/>  <PlayersNumberInGame xsi:nil=”true”/>  </game>  </gamesList>  <errorCode>0</errorCode>  <description/>  </response> |

* + 1. RegisterToken

Mount Point: ~/registerToken

The API is used to create a token for later on logging in the member to an Xpro.

The token created has a lifetime of two minutes before it expires.

A token can only be used once in order to login into a service.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| operatorId | mandatory | int | gspGameCurrencyId |
| username | mandatory | string | gspMember id  Has to be unique for operator. |
| accessPassword | mandatory | string | validation key  ex) privateKey: opToGspCurrency password  urlParam = “operatorId=” + gspGameCurrencyId  urlParam += “&username=” gspMemberId  urlParam += “&limitSetid=” limit set id of GsetXproGameList()  accessPassword = *MD5*(privateKey + urlParam) |
| props | mandatory | string | represents additional data to be stored for the token  additional data format: param1:value1;param2:valu2  For Mini Games: props should be empty |

Properties

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| limitsetid | mandatory | int | id of limit set that will be selected when using this registered token.(as given on GetGameListWithLimits) |
| ExternalSessionID | optional | string | session id of the external system  in case there is no ExternalSessionID, just with empty value after the colon |
| IsSuperSix | optional | bit | applies to Baccart only.  allows registering a token for invoking a Baccarat game in “Super Six” mode.  0: default mode  1: super six mode |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errorCode | mandatory | int | error code |
| description | mandatory | string | contain the login token in case no error occurred(0)  The error description if applicable to the error code supplied. |

Example

|  |
| --- |
| <response xmlns=”apiResultData” xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance” xmlns:xsd=”http://www.w3.org/2001/XMLSchema”>  <errorCode>0</errorCode>  <description>628ba503-85a6-4e3c-b909-7b845c0aae4c</description>  </response> |

|  |  |  |
| --- | --- | --- |
| **ERROR CODE** | **ERROR Name** | **DESCRIPTION** |
| 0 | NoError | Successful call |
| 2 | PlayerNotFound | No player with the sent username found |
| 5 | RequiredFieldMissing | Obligatory field missing |
| 6 | AuthenticationFailed | Sent access password does not match the server |
| 7 | ErrorCreatingToken | Unexpected error occurred or the token already exists (a retry might help) |
| 100 | UnspecifiedError | Unknown error |

* + 1. GetExternalGamesList, GetExternalGameURL (Unused)

Betsoft: used betsoft game in W88

MicroMartets: unused

* + 1. GetLastWinners

Mount Point: ~/getLastWinners

This methods returns list of the last winnings for the players of specific operator, from the last days specified.

The last winners will include the biggest win for each player that won in the requested date range.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| operatorId | mandatory | int | gspGameCurrencyId |
| accessPassword | mandatory | string | validation key  ex) privateKey: opToGspCurrency password  urlParam = “operatorId=” + gspGameCurrencyId  urlParam += “&username=” gspMemberId  urlParam += “&limitSetid=” limit set id of GsetXproGameList()  accessPassword = *MD5*(privateKey + urlParam) |
| daysBack | mandatory | int | The number of days back from which the winners will be fetched. |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errorCode | mandatory | int | error code |
| description | mandatory | string | contain the login token in case no error occurred(0)  The error description if applicable to the error code supplied. |
| winnsersList | mandatory | list<Winner> | container tag for all winners.  Empty when no winners are found. |

Winner

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| username | mandatory | string | the username of the winner |
| firstname | mandatory | string | first name of the winner |
| lastname | mandatory | string | last name initial of the winner |
| prize | mandatory | decimal | the amount won by the winner |

Example

|  |
| --- |
| <response xmlns="apiWinnersData" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <errorCode>0</errorCode>  <description />  <winnersList>  <winner>  <username>doni12</username>  <firstName>doni</firstName>  <lastName>d</lastName>  <prize>4500.0000</prize>  </winner>  <winner>  <username>Bob65</username>  <firstName>Jay</firstName>  <lastName>Z</lastName>  <prize>20.0000</prize>  </winner> </winnersList>  </response> |

|  |  |  |
| --- | --- | --- |
| **ERROR CODE** | **ERROR Name** | **DESCRIPTION** |
| 0 | NoError | Successful call |
| 5 | RequiredFieldMissing | Obligatory field missing |
| 6 | AuthenticationFailed | Sent access password does not match the server |
| 100 | UnspecifiedError | Unknown error |

* + 1. GetOperatorActivityReport (Unused)

This method returns a list of Roulette, Blackjack, Baccarat,Single Poker and Dragon-Tiger betting activity for certain operator in given date rangeincluding only closed rounds.

Note:Cancelled rounds will appear in according to cancelation date. The time range would not exceed 5hours and up to 1 month back.

* + 1. GetPlayersRejectedBets

This method returns a list players rejected bets per operator.

The time range would not exceed 1 day and up to 2 month back.

* + 1. GetPlayersChatReport (Unused)

This method returns a list of messages sent between players and dealers.

The method gets messages for certain operator in given date range.

Note: The time range would not exceed 2 days and up to 1 month back.

* + 1. GetPlayersRevenue

This method returns a list of messages sent between players and dealers.

The method gets messages for certain operator in given date range.

Note: The time range would not exceed 2 days and up to 1 month back.

* + 1. PlayerGetBalance (POST) - From XPro to IMS

Mount Point: ~/playerGetBalance

This API is used to get the specified player’s current balance given the player’s login.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Login | mandatory | string | player login?  CreateAccount 의 username |
| Session | optional | string | access token  4.7.7 RegisterToken():ExternalSessionID of props |
| OperatorId | optional | int | gspGameCurrencyId |

Example

|  |
| --- |
| {"Login":"nickname", "Session":"465663"} |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Data | mandatory | decimal | player balance  decimal with 4 digits after the decimal point |
| ErrorCode | mandatory | int | error code |
| HasErrors | mandatory | Boolean | true: has errors  false: no error |
| Message | mandatory | string | error message |

Example

|  |
| --- |
| {"d":{"Data":[10.5],"ErrorCode":0,"HasErrors":false,"Message":""}} |

|  |  |  |
| --- | --- | --- |
| **ERROR CODE** | **ERROR Name** | **DESCRIPTION** |
| 0 | NoError | Successful call |
| -1 | UnspecifiedError | Unknown error |
| -2 | RequiredFieldMissing | Obligatory field missing |
| -10 | InvalidPlayer | Player not found in Partner system |

* + 1. PlayerGetBalances (POST) - From XPro to IMS

Mount Point: ~/playerGetBalances

This service is used to get the get multiple player balances given their logins.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Logins | mandatory | List<string> | player login?  CreateAccount 의 username |
| OperatorId | optional | int | gspGameCurrencyId |

Example

|  |
| --- |
| {"Logins":["nickname1"," nickname2"]} |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Data | mandatory | List<Balance> | player balance  decimal with 4 digits after the decimal point |
| ErrorCode | mandatory | int | error code |
| HasErrors | mandatory | Boolean | true: has errors  false: no error |
| Message | mandatory | string | error message |

Balance

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Balance | mandatory | decimal | player balance  decimal with 4 digits after the decimal point |
| Login | mandatory | string | gspMember id |

Example

|  |
| --- |
| {"d":{"Data":[10.5],"ErrorCode":0,"HasErrors":false,"Message":""}} |

|  |  |  |
| --- | --- | --- |
| **ERROR CODE** | **ERROR Name** | **DESCRIPTION** |
| 0 | NoError | Successful call |
| -1 | UnspecifiedError | Unknown error |
| -2 | RequiredFieldMissing | Obligatory field missing |
| -10 | InvalidPlayer | Player not found in Partner system |

* + 1. Debit (POST) - From XPro to IMS

Mount Point: ~/Debit

This service is used to debit transfer.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Login | mandatory | string | player login?  CreateAccount 의 username |
| Session | optional | int | access token  4.7.7 RegisterToken():ExternalSessionID of props |
| GameId | mandatory | Int64 | Xpro game instance id |
| RoundId | mandatory | Int64 | XPG round (Spin) instance id |
| Sequence | mandatory | int | Debit transaction order number in a XPG Round.  (For tips transactions, sequence start from 1001) |
| Amount | mandatory | decimal | amount of executed transaction |
| DebitDetails | mandatory | string | <optional> – Bet: Betting Details  Sending of this data is per client requirement. |
| OperatorId | optional | int | gspGameCurrencyId |

Bet Detail

|  |  |
| --- | --- |
| **Game Type** | **Debit Details** |
| Baccarat | [GameType];[Bet\_Type\_ID]:[Bet\_Amount];….;[Bet\_Type\_ID]:[Bet\_Amount]  GameType: BC  BetTypes:  1 - Banker  2 - Player  3 - Tie  10 - PlayerPair  11 - BankerPair  13 - SuperSix  14 - Dragon  15 - Tiger  Example: “BC;2:25;11:30”  This parameter value will represent that player bet on Player with amount 25 and on Side bet BankerPair with amount 30. |
| SicBo | [GameType];[BetPointId]:[BetAmount];[BetPointId]:[BetAmount]  GameType: SB  “SB;6:25;33:30”  This example means that player bet on ThreeOfAKind4 with amount 25 and Sum16 with amount 30. |
| Roulette | [GameType];[BetPointId]:[BetAmount];[BetPointId]:[BetAmount]  GameType: RO  “RO;2:25;68:30”  This example means that player bet on 2 Straight Up with amount 25 and 17/20 Split with amount 30.  The whole list of bet point ids and their names can be found in the attached Excel spreadsheet. |
| BlackJack | [GameType];[Bet\_Type\_ID]:[Bet\_Amount];[Seat\_Index];[Hand\_Index]  GameType: BJ  Bet Type ID:  1 – Confirm seat bet  2 – Insurance Bet  3 – Split Bet  4 – DoubleDown bet  5 – BetBehind  ex) BJ;1:30;4;1  This example means that player placed a seated bet on seat number 4 and seat has one hand yet. |
| DragonTiger | [GameType];[Bet\_Type\_ID]:[Bet\_Amount];….;[Bet\_Type\_ID]:[Bet\_Amount]  GameType: DT  General bet types:  1 – Tiger  2 – Dragon  3 – Tie  Side Bet types:  17 - DragonBig  18 - DragonSmall  19 - TigerBig  20 - TigerSmall  21 - TigerDiamond  22 - TigerSpades  23 - TigerHearts  24 - TigerClubs  25 - DragonDiamond  26 - DragonSpades  27 - DragonHearts  28 – DragonClubs  ex) DT;3:25;23:30  This parameter value will represent that player bet on Tie with amount 25 and on Side bet TigerHearts with amount 30. |
| Single Poker | [GameType];[Bet\_Type\_ID]:[Bet\_Amount]  GameType: SP  [Bet\_Type\_ID] – is one from following list:  0 – Ante  1 – Flop  2 – Turn  3 – River  4 - Bonus  ex) SP;0:50;4:75  In this example Player put a bet on **Ante** with amount 50 and put a **Bonus** bet with amount 75.  Bonus bet can be missing – in this case parameter value will contain only one pair of Bet\_Type\_ID and Amount.  There will be also only one pair for other than Ante bet types. |
| Caribbean Poker | [GameType];[Bet\_Type\_ID]:[Bet\_Amount];[HandIndex]  GameType: CP  Bet Type ID:  1 – Ante bet  2 – Raise Bet  3 – Jackpot Bet  4 – Bet Behind  ex) “CP;4:30;4”  This example means that player placed a bet behind on seat number 4. |

Example

|  |
| --- |
| {"Login":"nickname", "Session": "455653", "GameId":4,"RoundId":236459, "Sequence":1, "Amount":50 } |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| ErrorCode | mandatory | int | error code |
| HasErrors | mandatory | Boolean | true: has errors  false: no error |
| Message | mandatory | string | error message |

Example

|  |
| --- |
| {"d":{"ErrorCode":0,"HasErrors":false,"Message":""}} |

|  |  |  |
| --- | --- | --- |
| **ERROR CODE** | **ERROR Name** | **DESCRIPTION** |
| 21 | DuplicateTransaction | Transaction has already been recorded in Partner server |
| 0 | NoError | Successful call |
| -1 | UnspecifiedError | Unknown error |
| -2 | RequiredFieldMissing | Obligatory field missing |
| -3 | DataOutOfRange | Field contains invalid or out of range data |
| -10 | InvalidPlayer | Player not found in Partner system |
| -11 | PlayerNotLoggedIn | Player not logged into Partner casino |
| -12 | PlayerSuspended | Player suspended by Partner casino operator |
| -20 | InsufficientFunds | Player balance is insufficient to place bet |
| -22 | AlteredTransaction | Transaction for specified RoundID and Nickname already recorded for different amount. |

* + 1. Credit (POST) - From XPro to IMS

Mount Point: ~/Credit

This service is used to Credit Transfer.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Login | mandatory | string | player login?  CreateAccount 의 username |
| Session | optional | int | access token  4.7.7 RegisterToken():ExternalSessionID of props |
| GameId | mandatory | Int64 | Xpro game instance id |
| RoundId | mandatory | Int64 | XPG round (Spin) instance id |
| Sequence | mandatory | int | Debit transaction order number in a XPG Round. (For tips transactions, sequence start from 1001) |
| Amount | mandatory | decimal | amount of executed transaction |
| CreditDetails | optional | string | details abount credit transaction  Sending of this data is per client requirement. |
| OperatorId | optional | int | gspGameCurrencyId |

Credit Detail

|  |  |
| --- | --- |
| **Game Type** | **Debit Details** |
| Baccarat | [GameType];[BankerCards]:[PlayerCards];[BetDetails] :…: [BetDetails]  GameType: BC  [GameType] – static value ‘BC’ for Baccarat  [BankerCards] – Banker’s cards codes as a comma separated list;  [PlayerCards] – Player’s cards codes as a comma separated list;  [BetDetails] – Comma separated list which includes betid, bettypeid, betamount, prizeamount;  For example: “BC;17,16:24,45;14184283,1, 15.00, 29.25”  This example shows that Banker cards were  and player cards were  BETID on XPRO side was 14184283  Player bet on Banker with amount of 15.00 and won 29.25 which means that Banker commission was 0.75 . |
| SicBo | [GameType];[BetPointId]:[BetAmount];[BetPointId]:[BetAmount]  GameType: SB  [Result] – it represents comma separated list of 3 numbers  [BetDetails] – comma separated list of BetPointId, BetAmount and Prize without spaces.  For example:  “SB;3,1,2;16,1.00,2.00:17,1.00,2.00:28,1.00,0.00:33,1.00,0.00:35,1.00,6.00:”  This example provides information about the round that result was 3, 1, 2    Player bet on OneOfAKind2 with 1 and won 2, OneOfAKind3 with 1 and won 2, on Sum11 with 1 and lost, on Sum16 with 1 and lost and on Duo12 with 1 and had prize 6. |
| Roulette | [GameType];[BetPointId]:[BetAmount];[BetPointId]:[BetAmount]  GameType: RO  [Result] – it represents just the winning number for Roulette;  [BetDetails] – comma separated list of BetPointId, BetAmount and Prize without spaces.  For example:  “RO;4;88,70,0: 5,70,0:30,70,0:26,70,0:18,70,0:17,70,0:11,70,0”  This example means that for current round there was 4 as winning number and player bet with 70 on 7 different bet points (5 Straight Up, 11 Straight Up, 17 Straight Up, 18 Straight Up, 26 Straight Up, 30 Straight Up, 29/32 Split) |
| BlackJack | [GameType];[Bet\_Type\_ID]:[Bet\_Amount];[Seat\_Index];[Hand\_Index]  GameType: BJ  [DealerCards] – Dealer’s cards codes as a comma separated list;  [PlayerCards] – comma separated list of pairs with player’s card code and hand id separated by $;  [BetDetails] – Comma separated list which includes betid, bettypeid, handid, betamount, prizeamount without spaces;  “BJ;42,16,28,46,21:25$3534300,49$3534300,11$3534299,34$3534299,30$3534301,1$3534301;2788778,1,3534300,25.00,50.00: 2788779,1,3534299,25.00,50.00: 2788780,1,3534301,25.00,50.00;”  This example says that dealer cards were  Player cards were on hand 3534300  on hand 3534299  on hand 3534301  Also players had 3 bets of type 1 – Confirm seat bet (see debit details for BlackJack) with amount of 25 and he won on each of 3 hands 50. |
| DragonTiger | [GameType];[Bet\_Type\_ID]:[Bet\_Amount];….;[Bet\_Type\_ID]:[Bet\_Amount]  GameType: DT  [DragonCard] – code of the card dealt for Dragon  [TigerCard] - code of the card dealt for Tiger  [BetDetails] – Comma separated list which includes betid, bettypeid, betamount, prizeamount without spaces.  For example:  “DT;47:5;4042665,3,100.00,0.00”  Provided example shows that card dealt for Dragon was and for Tiger was .  Player bet on Tie with amount of 100 and lost. |
| Single Poker | [GameType];[Bet\_Type\_ID]:[Bet\_Amount]  GameType: SP  [PlayerCards] – commaseparated list with codes of cards dealt to the Player;  [DealerCards] – commaseparated list with codes of cards dealt to the Dealer;  [CommunityCards] – commaseparated list with codes of cards dealt to the Community(Flop, Turn, River cards)  [BetDetails] – AnteBetAmount, FlopBetAmount,TurnBetAmount, RiverBet, BetPrize separated by comma;  [BonusBetDetails] – Bonus bet and bonus bet prize separated by comma.  For example:  “SP;29,41:24,18:48,47,6,7,28;100.00,200.00,0.00,0.00,0.00;100.00, 0.00”  This means that Player cards were  , Dealer cards were  and Community cards were  Player bet on ante with 100, on Flop he/she placed 2\* ante which is 200 and also he/she placed a bonus bet with 100 and he/she lost all bets. |
| Caribbean Poker | [GameType];[Bet\_Type\_ID]:[Bet\_Amount];[HandIndex]  GameType: CP  [DealerCards] - comma separated list of codes of cards dealt to Player  [PlayerCards] – comma separated list of codes of cards dealt to Player  [BetDetails] – bet id, hand index, bet type, Ante bet amount, Raise bet amount, prize separated by comma without spaces.  For example:  “CP;24,52,45,3,27:42,25,36,29,4;119872,1,1,100.00,200.00:119873,1,2,200.00,400.00”  It says that Dealer cards were     , Player cards -  Player placed 100 as Ante bet and respectively 200 as Raise bet and won 200 for Ante and 400 for Raise. |

Example

|  |
| --- |
| {"Login":"nickname", "Session": "455653", "GameId":4,"RoundId":236459, "Amount":10 } |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| ErrorCode | mandatory | int | error code |
| HasErrors | mandatory | Boolean | true: has errors  false: no error |
| Message | mandatory | string | error message |

Example

|  |
| --- |
| {"d":{"ErrorCode":0,"HasErrors":false,"Message":""}} |

|  |  |  |
| --- | --- | --- |
| **ERROR CODE** | **ERROR Name** | **DESCRIPTION** |
| 21 | DuplicateTransaction | Transaction has already been recorded in Partner server |
| 0 | NoError | Successful call |
| -1 | UnspecifiedError | Unknown error |
| -2 | RequiredFieldMissing | Obligatory field missing |
| -10 | InvalidPlayer | Player not found in Partner system |
| -11 | PlayerNotLoggedIn | Player not logged into Partner casino |
| -12 | PlayerSuspended | Player suspended by Partner casino operator |
| -22 | AlteredTransaction | Transaction for specified RoundID and Nickname already recorded for different amount. |
| -30 | InvalidGame | No game recode exists for this game |

* + 1. CancelTransaction (POST) - From XPro to IMS

Mount Point: ~/CancelTransaction

This method will cancel transaction for incomplete game or for the debit transaction that didn’t received a response from partner server in a reasonable time in order to ensure that the failed debit transaction was not processed on the partner side.

CancelTransaction will have up to 3 retry attempts in case there is no response from partner server

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Login | mandatory | string | player login?  CreateAccount 의 username |
| Session | optional | int | access token  4.7.7 RegisterToken():ExternalSessionID of props |
| GameId | mandatory | Int64 | Xpro game instance id |
| RoundId | mandatory | Int64 | XPG round (Spin) instance id |
| Sequence | mandatory | int | Debit transaction order number in a XPG Round.  (For tips transactions, sequence start from 1001) |
| OperatorId | optional | int | gspGameCurrencyId |

Example

|  |
| --- |
| {"Login":"nickname","Session":"345346","GameId":1,"RoundId":240026,"Sequence":1} |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| ErrorCode | mandatory | int | error code |
| HasErrors | mandatory | Boolean | true: has errors  false: no error |
| Message | mandatory | string | error message |

Example

|  |
| --- |
| {"d":{"ErrorCode":0,"HasErrors":false,"Message":""}} |

|  |  |  |
| --- | --- | --- |
| **ERROR CODE** | **ERROR Name** | **DESCRIPTION** |
| 0 | NoError | Successful call |
| -1 | UnspecifiedError | Unknown error |
| -2 | RequiredFieldMissing | Obligatory field missing |
| -10 | InvalidPlayer | Player not found in Partner system |
| -30 | InvalidGame | No game recode exists for this game |

* + 1. CancelRound (POST) - From XPro to IMS

Mount Point: ~/CancelRound

This method will cancel a complete round in a specified game that should issue on partner side “transferring” the bets back to the player’s balance and removing any payouts.

CancelRound will be resent for 3 times in case there will be timeout for the method request.

InvalidGame would be an expected response for a resubmitted CancelRound. This error code could be considered a ‘success’ since the game does not exist then it is functionally cancelled

Tips transactions for such a round should not be cancelled. Tips sequences will start from 1001.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Logins | mandatory | List<string> | List of player login?  CreateAccount 의 username |
| GameId | mandatory | Int64 | Xpro game instance id |
| RoundId | mandatory | Int64 | XPG round (Spin) instance id |
| OperatorId | optional | int | gspGameCurrencyId |

Example

|  |
| --- |
| {"GameId":4,"RoundId":236539,"Logins":["nickname1"," nickname2"]} |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| ErrorCode | mandatory | int | error code |
| HasErrors | mandatory | Boolean | true: has errors  false: no error |
| Message | mandatory | string | error message |

Example

|  |
| --- |
| {"d":{"ErrorCode":0,"HasErrors":false,"Message":""}} |

|  |  |  |
| --- | --- | --- |
| **ERROR CODE** | **ERROR Name** | **DESCRIPTION** |
| 0 | NoError | Successful call |
| -1 | UnspecifiedError | Unknown error |
| -2 | RequiredFieldMissing | Obligatory field missing |
| -30 | InvalidGame | No game recode exists for this game |

* + 1. GetBetHistory

Mount Point: ~/getBetHistory

The API is used to get a list of Roultte, Blackjack, Baccarat, Single Poker and Dragon-Tiger betting activity for certain operator in given dage range including only closed rounds.

Note: Cancelled rounds will appear in according to cancelation date. The time range would not exceed 5hours and up to 1 month back.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| operatorId | mandatory | int | gspGameCurrencyId |
| accessPassword | mandatory | string | validation key  ex) privateKey: Branch Password  urlParam = “operatorId=” + game.getBranchID();  urlParam += “&fromDate=” + startDate;  urlParam += “&toDate=” + endDate;  urlParam += “&getBalanceDetails=1”;  urlParam += “&getExternalGamesActivity=1”;  accessPassword = *MD5*(privateKey + urlParam) |
| fromDate | mandatory | datetime | query start date |
| toDate | optional | datetime | query end date |
| getExternalGamesActivity | optional | Boolean | Whether to include Casino games (Not Live) information.  Possible values are (0, 1).  Default value is 0 in case parameter is not sent. |
| getTipActivity | optional | Boolean | Whether to include tips information given by players. Default value is 0 in case parameter is not sent. |
| getBalanceDetails | optional | Boolean | Whether to include balance information for betting activity report. Possible values are (0,1). Default value is 0 in case parameter is not sent. |
| includeBJBetType | optional | Boolean | Whether to include bet type information in Black Jack Bets section.  Possible values are (0, 1).  Default value is 1 in case parameter is not sent. |

#### Response

The maximum date will be Now minus 5 min in case toDate is empty.if a larger date will be received it will be adjusted and the error code for this call will be 23.

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errorCode | mandatory | int | error code |
| description | mandatory | string | contain the login token in case no error occurred(0)  the error description if applicable to the error code supplied. |
| externalRecordID | mandatory | string | identifier for the date of the fetch.  Can be used for the next report call. |
| CompressedData | mandatory | string | This is a container for the bets history data.  The data is compressed with zip, in order to prevent transmission of big messages.  decompression process is needed. |

Example

|  |
| --- |
| <response xmlns=”apiOperatorActivityReport” xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance” xmlns:xsd=”http://www.w3.org/2001/XMLSchema”>  <errorCode>0</errorCode>  <description/>  <externalRecordID>634248568267130000</externalRecordID>  <CompressedData length=”156324”>  ……  </CompressedData>  </response> |

Decompressed Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| rounds | mandatory | List<XproRound> |  |
| rouletteBets | mandatory | List<XproRouletteBet> |  |
| baccaratBets | mandatory | List<XproBaccaratBet> |  |
| baccaratHands | mandatory | List<XproBaccaratHand> |  |
| carribeanPokerBets | mandatory | List<XproCarribeanPokerBet> |  |
| carribeanPokerHands | mandatory | List<XproCarribeanPokerHands> |  |
| blackjackBets | mandatory | List<XproBlackjackBets> |  |
| blackjackHands | mandatory | List<XproBlackjackHand> |  |
| dragonTigerBets | mandatory | List<XproDragonTigerBet> |  |
| dragonTigerHands | mandatory | List<XproDragonTigerHand> |  |
| singlePokerBets | mandatory | List<XproSinglePokerBet> |  |
| singlePokerHands | mandatory | List<XproSinglePokerHand> |  |
| sicBoBets | mandatory | List<XproSicBoBet > |  |
| ExternalGamesActivity | mandatory | List<XproExternalGamesActivity > |  |
| TipsActivity | mandatory | List<XproTipsActivity> |  |

XproRound

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| roundID | mandatory | string | round id?  round definition? |
| gameType | mandatory | string | game type |
| gameName | mandatory | string | game name |
| roundDate | mandatory | string | round date |
| roundResult | mandatory | string | round result?  winloss amount? |
| totalBets | mandatory | decimal | total bet amount |
| totalPrizes | mandatory | decimal | total prize amount |

XproRouletteBet

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| roundID | mandatory | string | round id?  round definition? |
| username | mandatory | string | gspMember id |
| betPointID | mandatory | string | bet point id? |
| betType | mandatory | string | bet type |
| betAmount | mandatory | decimal | bet amount |
| prizeAmount | mandatory | decimal | prize amount |
| startBalance | mandatory | decimal | balance before bet |
| endBalance | mandatory | decimal | balance after bet |

XproBaccaratBet

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| roundID | mandatory | string | round id?  round definition? |
| username | mandatory | string | gspMember id |
| betID | mandatory | string | bet id? |
| betType | mandatory | string | bet type |
| betAmount | mandatory | decimal | bet amount |
| prizeAmount | mandatory | decimal | prize amount |
| isSuperSix | mandatory | string | 0: default mode  1: super six mode |
| startBalance | mandatory | decimal | balance before bet |
| endBalance | mandatory | decimal | balance after bet |

XproBaccaratHand

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| roundID | mandatory | string | round id?  round definition? |
| handType | mandatory | string | hand type? |
| cardSeq | mandatory | string | card sequence? |
| cardID | mandatory | string | card id  card definition |

XproCarribeanPokerBet

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| roundID | mandatory | string | round id?  round definition? |
| username | mandatory | string | gspMember id |
| seatNumber | mandatory | string | seat number |
| betType | mandatory | string | bet type |
| anteBetAmount | mandatory | decimal | ante bet amount |
| raiseBetAmount | mandatory | decimal | raise bet amount |
| betPrize | mandatory | decimal | prize amount |
| startBalance | mandatory | decimal | balance before bet |
| endBalance | mandatory | decimal | balance after bet |

XproCarribeanPokerHands

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| roundID | mandatory | string | round id?  round definition? |
| seatNumber | mandatory | string | seat number |
| card1 | mandatory | string | card 1 |
| card2 | mandatory | string | card 2 |
| card3 | mandatory | string | card 3 |
| card4 | mandatory | string | card 4 |
| card5 | mandatory | string | card 5 |
| handValue | mandatory | string | hand value |
| winningHand | mandatory | string | winning hand |

XproBlackjackBets

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| roundID | mandatory | string | round id?  round definition? |
| username | mandatory | string | gspMember id |
| handID | mandatory | string | hand id |
| betID | mandatory | string | bet id |
| betType | mandatory | string | betType |
| betAmount | mandatory | decimal | bet amount |
| prizeAmount | mandatory | decimal | prize amount |
| startBalance | mandatory | decimal | balance before bet |
| endBalance | mandatory | decimal | balance after bet |

XproBlackjackHand

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| roundID | mandatory | string | round id?  round definition? |
| handID | mandatory | string | hand id |
| isDealerHand | mandatory | string | dealer hand? |
| cardSeq | mandatory | string | card sequence number |
| cardID | mandatory | string | card id |

XproDragonTigerBet

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| roundID | mandatory | string | round id?  round definition? |
| username | mandatory | string | gspMember id |
| betID | mandatory | string | bet id |
| betType | mandatory | string | betType |
| betAmount | mandatory | decimal | bet amount |
| prizeAmount | mandatory | decimal | prize amount |
| startBalance | mandatory | decimal | balance before bet |
| endBalance | mandatory | decimal | balance after bet |

XproDragonTigerHand

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| roundID | mandatory | string | round id?  round definition? |
| handType | mandatory | string | hand type |
| cardSeq | mandatory | string | card sequence number |
| cardID | mandatory | string | card id |

XproSinglePokerBet

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| roundID | mandatory | string | round id?  round definition? |
| username | mandatory | string | gspMember id |
| anteBetAmount | mandatory | decimal | ante bet amount |
| flopBetAmount | mandatory | decimal | flop bet amount |
| anteBetAmount | mandatory | decimal | ante bet amount |
| turnBetAmount | mandatory | decimal | turn bet amount |
| riverBetAmount | mandatory | decimal | river bet amount |
| betPrize | mandatory | decimal | prize amount |
| bonusBetAmount | mandatory | decimal | bonus bet amount  bonus definition |
| bonusPrize | mandatory | decimal | bonus prize amount |
| startBalance | mandatory | decimal | balance before bet |
| endBalance | mandatory | decimal | balance after bet |

XproSinglePokerHand

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| roundID | mandatory | string | round id?  round definition? |
| playerCard1 | mandatory | string | player card 1 |
| playerCard2 | mandatory | string | player card 2 |
| dealerCard1 | mandatory | string | dealer card 2 |
| dealerCard2 | mandatory | string | dealer card 2 |
| communityCard1 | mandatory | string | community card 1 |
| communityCard2 | mandatory | string | community card 2 |
| communityCard3 | mandatory | string | community card 3 |
| communityCard4 | mandatory | string | community card 4 |
| communityCard5 | mandatory | string | community card 5 |
| playerHand | mandatory | string | player hand |
| dealerHand | mandatory | string | dealer hand |
| winningHand | mandatory | string | winning hand |

XproSicBoBet

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| roundID | mandatory | string | round id?  round definition? |
| username | mandatory | string | gspMember id |
| betPointID | mandatory | string | bet point id |
| betName | mandatory | string | bet name |
| betAmount | mandatory | decimal | bet amount |
| betPrize | mandatory | decimal | prize amount |
| startBalance | mandatory | decimal | balance before bet |
| endBalance | mandatory | decimal | balance after bet |

XproExternalGamesActivity

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| roundID | mandatory | string | round id?  round definition? |
| nickname | mandatory | string | gspMember id |
| gameName | mandatory | string | game name |
| amount | mandatory | decimal | bet amount? |
| prize | mandatory | decimal | prize amount |
| updateDate | mandatory | string | update date |

XproTipsActivity

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| roundID | mandatory | string | round id?  round definition? |
| nickname | mandatory | string | gspMember id |
| gameName | mandatory | string | game name |
| amount | mandatory | decimal | bet amount? |
| transactionDate | mandatory | string | transaction date |

Decompressed Response Example

|  |
| --- |
| <Data>  <Rounds>  <RoundID>1717276</RoundID>  <GameType>Blackjack</GameType>  <GameName>Test Blackjack 1 PP</GameName>  <RoundDate>2010-11-08T20:54:06.33+01:00</RoundDate>  <RoundResult>-1</RoundResult>  <TotalBets>25.0000</TotalBets>  <TotalPrizes>0.0000</TotalPrizes>  </Rounds>  <Rounds>  <RoundID>1718790</RoundID>  <GameType>Baccarat</GameType>  <GameName>Test Baccarat 1 PP</GameName>  <RoundDate>2010-11-09T09:36:19.483+01:00</RoundDate>  <RoundResult>1</RoundResult>  <TotalBets>20.0000</TotalBets>  <TotalPrizes>19.5000</TotalPrizes>  </Rounds>  <Rounds>  <RoundID>15074842</RoundID>  <GameType>Caribbean Poker</GameType>  <GameName>first CP</GameName>  <RoundDate>2013-12-04T11:45:26.06+02:00</RoundDate>  <RoundResult>-1</RoundResult>  <TotalBets>10.0000</TotalBets>  <TotalPrizes>0.0000</TotalPrizes>  </Rounds>  <Rounds>  <RoundID>1718796</RoundID>  <GameType>Roulette</GameType>  <GameName>Europe Live 1 PP</GameName>  <RoundDate>2010-11-09T09:37:57.627+01:00</RoundDate>  <RoundResult>25</RoundResult>  <TotalBets>40.0000</TotalBets>  <TotalPrizes>180.0000</TotalPrizes>  </Rounds>  <Rounds>  <RoundID>8212598</RoundID>  <GameType>DragonTiger</GameType>  <GameName>Dragon Tiger 1</GameName>  <RoundDate>2012-11-20T11:07:45.927+01:00</RoundDate>  <RoundResult>2</RoundResult>  <TotalBets>175.0000</TotalBets>  <TotalPrizes>250.0000</TotalPrizes>  </Rounds>  <Rounds>  <RoundID>1718831</RoundID>  <GameType>SicBo</GameType>  <GameName>Live SicBo 1</GameName>  <RoundDate>2013-01-16T10:04:12.326+01:00</RoundDate>  <RoundResult>0</RoundResult>  <TotalBets>800.0000</TotalBets>  <TotalPrizes>400.0000</TotalPrizes>  </Rounds>  <RouletteBets>  <RoundID>1718796</RoundID>  <Username>qa06</Username>  <BetPointID>89</BetPointID>  <BetAmount>10.0000</BetAmount>  <PrizeAmount>0.0000</PrizeAmount>  </RouletteBets>  <RouletteBets>  <RoundID>1718796</RoundID>  <Username>qa06</Username>  <BetPointID>80</BetPointID>  <BetAmount>10.0000</BetAmount>  <PrizeAmount>180.0000</PrizeAmount>  </RouletteBets>  <RouletteBets>  <RoundID>1718796</RoundID>  <Username>qa06</Username>  <BetPointID>65</BetPointID>  <BetAmount>10.0000</BetAmount>  <PrizeAmount>0.0000</PrizeAmount>  </RouletteBets>  <RouletteBets>  <RoundID>1718796</RoundID>  <Username>qa06</Username>  <BetPointID>59</BetPointID>  <BetAmount>10.0000</BetAmount>  <PrizeAmount>0.0000</PrizeAmount>  </RouletteBets>  <CarribeanPokerBets>  <RoundID>15074842</RoundID>  <Username>sasha</Username>  <SeatNumber>2</SeatNumber>  <BetType>Seated Bet</BetType>  <AnteBetAmount>10.0000</AnteBetAmount>  <RaiseBetAmount>0.0000</RaiseBetAmount>  <BetPrize>0.0000</BetPrize>  </CarribeanPokerBets>  <CarribeanPokerHands>  <RoundID>15074842</RoundID>  <SeatNumber>-1</SeatNumber>  <Card1>28</Card1>  <Card2>25</Card2>  <Card3>36</Card3>  <Card4>11</Card4>  <Card5>44</Card5>  <HandValue>0</HandValue>  <WinningHand>0</WinningHand>  </CarribeanPokerHands>  <BlackjackBets>  <RoundID>1717276</RoundID>  <Username>qa06</Username>  <HandID>1504510</HandID>  <BetID>1144935</BetID>  <BetAmount>25.0000</BetAmount>  <BetType>Seated Bet</BetType>  <PrizeAmount>0.0000</PrizeAmount>  </BlackjackBets>  <BlackjackHands>  <RoundID>1717276</RoundID>  <HandID>1504511</HandID>  <IsDealerHand>1</IsDealerHand>  <CardSeq>0</CardSeq>  <CardID>9</CardID>  </BlackjackHands>  <BlackjackHands>  <RoundID>1717276</RoundID>  <HandID>1504511</HandID>  <IsDealerHand>1</IsDealerHand>  <CardSeq>1</CardSeq>  <CardID>40</CardID>  </BlackjackHands>  <BlackjackHands>  <RoundID>1717276</RoundID>  <HandID>1504510</HandID>  <IsDealerHand>0</IsDealerHand>  <CardSeq>0</CardSeq>  <CardID>4</CardID>  </BlackjackHands>  <BlackjackHands>  <RoundID>1717276</RoundID>  <HandID>1504510</HandID>  <IsDealerHand>0</IsDealerHand>  <CardSeq>1</CardSeq>  <CardID>25</CardID>  </BlackjackHands>  <BaccaratBets>  <RoundID>1718790</RoundID>  <Username>qa06</Username>  <BetID>657064</BetID>  <BetType>Banker</BetType>  <BetAmount>10.0000</BetAmount>  <PrizeAmount>19.5000</PrizeAmount>  <IsSuperSix>1</IsSuperSix>  </BaccaratBets>  <BaccaratBets>  <RoundID>1718790</RoundID>  <Username>qa06</Username>  <BetID>657065</BetID>  <BetType>Player</BetType>  <BetAmount>10.0000</BetAmount>  <PrizeAmount>0.0000</PrizeAmount>  <IsSuperSix>1</IsSuperSix>  </BaccaratBets>  <BaccaratHands>  <RoundID>1718790</RoundID>  <HandType>1</HandType>  <CardSeq>0</CardSeq>  <CardID>10</CardID>  </BaccaratHands>  <BaccaratHands>  <RoundID>1718790</RoundID>  <HandType>1</HandType>  <CardSeq>1</CardSeq>  <CardID>34</CardID>  </BaccaratHands>  <BaccaratHands>  <RoundID>1718790</RoundID>  <HandType>2</HandType>  <CardSeq>0</CardSeq>  <CardID>37</CardID>  </BaccaratHands>  <BaccaratHands>  <RoundID>1718790</RoundID>  <HandType>2</HandType>  <CardSeq>1</CardSeq>  <CardID>4</CardID>  </BaccaratHands>  <DragonTigerBets>  <RoundID>8212598</RoundID>  <Username>RoyKee</Username>  <BetID>2273883</BetID>  <BetType>Dragon</BetType>  <BetAmount>125.0000</BetAmount>  <PrizeAmount>250.0000</PrizeAmount>  </DragonTigerBets>  <DragonTigerBets>  <RoundID>8212598</RoundID>  <Username>RoyKee</Username>  <BetID>2273884</BetID>  <BetType>Tie</BetType>  <BetAmount>50.0000</BetAmount>  <PrizeAmount>0.0000</PrizeAmount>  </DragonTigerBets>  <DragonTigerHands>  <RoundID>8212598</RoundID>  <HandType>1</HandType>  <CardSeq>0</CardSeq>  <CardID>43</CardID>  </DragonTigerHands>  <DragonTigerHands>  <RoundID>8212598</RoundID>  <HandType>2</HandType>  <CardSeq>0</CardSeq>  <CardID>33</CardID>  </DragonTigerHands>  <SinglePokerBets>  <RoundID>177224</RoundID>  <Username>momi</Username>  <AnteBetAmount>174.0000</AnteBetAmount>  <FlopBetAmount>348.0000</FlopBetAmount>  <TurnBetAmount>0.0000</TurnBetAmount>  <RiverBetAmount>0.0000</RiverBetAmount>  <BetPrize>696.0000</BetPrize>  <BonusBetAmount>0.0000</BonusBetAmount>  <BonusPrize>0.0000</BonusPrize>  </SinglePokerBets>  <SinglePokerBets>  <RoundID>177225</RoundID>  <Username>momi</Username>  <AnteBetAmount>174.0000</AnteBetAmount>  <FlopBetAmount>0.0000</FlopBetAmount>  <TurnBetAmount>0.0000</TurnBetAmount>  <RiverBetAmount>0.0000</RiverBetAmount>  <BetPrize>0.0000</BetPrize>  <BonusBetAmount>0.0000</BonusBetAmount>  <BonusPrize>0.0000</BonusPrize>  </SinglePokerBets>  <SinglePokerBets>  <RoundID>177226</RoundID>  <Username>momi</Username>  <AnteBetAmount>174.0000</AnteBetAmount>  <FlopBetAmount>0.0000</FlopBetAmount>  <TurnBetAmount>0.0000</TurnBetAmount>  <RiverBetAmount>0.0000</RiverBetAmount>  <BetPrize>0.0000</BetPrize>  <BonusBetAmount>0.0000</BonusBetAmount>  <BonusPrize>0.0000</BonusPrize>  </SinglePokerBets>  <SinglePokerBets>  <RoundID>177227</RoundID>  <Username>momi</Username>  <AnteBetAmount>174.0000</AnteBetAmount>  <FlopBetAmount>348.0000</FlopBetAmount>  <TurnBetAmount>0.0000</TurnBetAmount>  <RiverBetAmount>0.0000</RiverBetAmount>  <BetPrize>696.0000</BetPrize>  <BonusBetAmount>0.0000</BonusBetAmount>  <BonusPrize>0.0000</BonusPrize>  </SinglePokerBets>  <SicBoBets>  <RoundID>1718831</RoundID>  <Username>TEST1407</Username>  <BetPointID>16</BetPointID>  <BetName>OneOfAKind2</BetName>  <BetAmount>200.0000</BetAmount>  <BetPrize>400.0000</BetPrize>  </SicBoBets>  <SicBoBets>  <RoundID>1718831</RoundID>  <Username>TEST1407</Username>  <BetPointID>18</BetPointID>  <BetName>OneOfAKind4</BetName>  <BetAmount>200.0000</BetAmount>  <BetPrize>0.0000</BetPrize>  </SicBoBets>  <SicBoBets>  <RoundID>1718831</RoundID>  <Username>TEST1407</Username>  <BetPointID>20</BetPointID>  <BetName>OneOfAKind6</BetName>  <BetAmount>200.0000</BetAmount>  <BetPrize>12.0000</BetPrize>  </SicBoBets>  <SicBoBets>  <RoundID>1718831</RoundID>  <Username>TEST1407</Username>  <BetPointID>28</BetPointID>  <BetName>Sum11</BetName>  <BetAmount>200.0000</BetAmount>  <BetPrize>0.0000</BetPrize>  </SicBoBets>  <SicBoResults>  <RoundID>1718831</RoundID>  <Result1>3</Result1>  <Result2>6</Result2>  <Result3>4</Result3>  </SicBoResults>  <SinglePokerHands>  <RoundID>177274</RoundID>  <PlayerCard1>17</PlayerCard1>  <playerCard2>16</playerCard2>  <DealerCard1>45</DealerCard1>  <DealerCard2>29</DealerCard2>  <CommunityCard1>9</CommunityCard1>  <CommunityCard2>11</CommunityCard2>  <CommunityCard3>24</CommunityCard3>  <CommunityCard4>44</CommunityCard4>  <CommunityCard5>43</CommunityCard5>  <PlayerHand>2</PlayerHand>  <DealerHand>1</DealerHand>  <WinningHand>2</WinningHand>  </SinglePokerHands>  <SinglePokerHands>  <RoundID>177333</RoundID>  <PlayerCard1>33</PlayerCard1>  <playerCard2>46</playerCard2>  <DealerCard1>32</DealerCard1>  <DealerCard2>1</DealerCard2>  <CommunityCard1>41</CommunityCard1>  <CommunityCard2>50</CommunityCard2>  <CommunityCard3>49</CommunityCard3>  <CommunityCard4>17</CommunityCard4>  <CommunityCard5>4</CommunityCard5>  <PlayerHand>2</PlayerHand>  <DealerHand>1</DealerHand>  <WinningHand>2</WinningHand>  </SinglePokerHands>  <SinglePokerHands>  <RoundID>177334</RoundID>  <PlayerCard1>34</PlayerCard1>  <playerCard2>26</playerCard2>  <DealerCard1>40</DealerCard1>  <DealerCard2>47</DealerCard2>  <CommunityCard1>7</CommunityCard1>  <CommunityCard2>22</CommunityCard2>  <CommunityCard3>20</CommunityCard3>  <CommunityCard4>42</CommunityCard4>  <CommunityCard5>8</CommunityCard5>  <PlayerHand>2</PlayerHand>  <DealerHand>2</DealerHand>  <WinningHand>1</WinningHand>  </SinglePokerHands>  <ExternalGamesActivity>  <NickName>bsgTest4</NickName>  <GameName>4\_247</GameName>  <Amount>0.7600</Amount>  <Prize>4.8000</Prize>  <UpdateDate>2012-11-13T15:54:25.51+02:00</UpdateDate>  <RoundId>89297659</RoundId>  </ExternalGamesActivity>  <TipsActivity>  <NickName>IDN197505</NickName>  <GameName>Dragon Tiger 1</GameName>  <RoundId>8131828</RoundId>  <Amount>30.0000</Amount>  <TransactionDate>2012-11-13T15:17:09.38+02:00  </TransactionDate>  </TipsActivity>  </Data> |

* 1. Laxino Gaming (Laxino) – TBD
  2. XTD (XTD)
     1. ValidateToken (POST)

**Mount Point:** /XTD/Player.svc/authenticate

This API is to validate the access token.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | access token used when launches the lobby or slot game. |

Example

|  |
| --- |
| {  "accessToken":"D1250CF0A6C849E983D0D61084D54C0B",  } |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| returnCode | mandatory | int | response return code |
| returnMessage | mandatory | string | error message |
| playerId | mandatory | string | player Id |
| accessToken | mandatory | string | access token |
| nickname | mandatory | string | nickname for display in the game |
| currencyIsoCd | mandatory | string | currency ISO code |
| balance | mandatory | long | balance ( in cents per currency)  KRW 100 means KRW 100  USD 100 means USD 1.00 |

Example

|  |
| --- |
| {  "returnCode": 0,  "returnMessage": null,  "playerId": "CO\_38895",  "accessToken": "D1250CF0A6C849E983D0D61084D54C0B",  "nickname": "thb4",  "currencyIsoCd": "THB",  "balance": 87000  } |

* + 1. GetBalance (GET)

**Mount Point:** /XTD/Player.svc/token/{ACCESSTOKEN}/balance

This API is to retrieve the player’s balance.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | token used when launches the lobby or slot game. |

Example

|  |
| --- |
| ~/XTD/Player.svc/token/E9BDD39F1F704E0C8477EA60E9C52C6B/balance |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| returnCode | mandatory | int | response return code |
| returnMessage | mandatory | string | error message |
| currencyIsoCd | mandatory | string | currency ISO code |
| balance | mandatory | long | balance ( in cents per currency)  KRW 100 means KRW 100  USD 100 means USD 1.00 |

Example

|  |
| --- |
| {  "returnCode": 0,  "returnMessage": null,  "currencyIsoCd": "THB",  "balance": 5000  } |

* + 1. PostTransfer (POST)

**Mount Point:** /XTD/Player.svc/postTransfer

This API is to be used when places a bet or win the game or cancel a bet.

This API should be called even payout(win) amount is zero.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| accessToken | mandatory | string | token used when launches the lobby or slot game. |
| playerId | mandatory | string | player id |
| transactionId | mandatory | string | unique transaction id in XTD system |
| currencyIsoCd | mandatory | string | player’s currency ISO code |
| amount | mandatory | string | transaction amount ( in cents )  Debit = negative balance  Credit = positive balance |
| gameType | mandatory | int | 10: Live Casino  20: Slots |
| transactionType | mandatory | int | 11005: Debit (Bet, Payout Cancel, Tip)  11006: Credit (Payout, Bet Cancel) |
| description | mandatory | int | transaction description  140001: Bet  140002: Bet Cancel  140003: Payout  140004: Payout Cancel  140005: Tip |
| gameId | mandatory | string | unique game id in the XTD |
| gameName | mandatory | string | game name in the XTD |
| roundId | mandatory | string | round id  Debit and Credit should be same round id |

Example

|  |
| --- |
| {  "accessToken":"4ED36F99AA8749C98937EBEAEF62C853",  "playerId":"CO\_38896",  "transactionId":"xtd\_transactionId\_1",  "currencyIsoCd":"THB",  "amount":100,  "gameType":10,  "transactionType":11006,  "description":140003,  "gameId":"1",  "gameName":"xtd test game",  "roundId":"xtdroundid\_1"  } |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| returnCode | mandatory | int | response return code |
| returnMessage | mandatory | string | error message |
| currencyIsoCd | mandatory | string | currency ISO code |
| balance | mandatory | long | balance ( in cents per currency)  KRW 100 means KRW 100  USD 100 means USD 1.00 |

Example

|  |
| --- |
| {  "returnCode": 0,  "returnMessage": null,  "currencyIsoCd": "THB",  "balance": 5100  } |

* + 1. RollbackTransfer (POST)

**Mount Point:** /XTD/Player.svc/rollback

This API is to used when XTD didn’t get the response of PostTransfer(Debit) due to network system issue or timeout.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| playerId | mandatory | string | player id |
| transactionId | mandatory | string | transaction id wish to rollback |

Example

|  |
| --- |
| {  "playerId":" CO\_38896",  "transactionId":" xtd\_transactionId\_1"  } |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| returnCode | mandatory | int | response return code |
| returnMessage | mandatory | string | error message |
| currencyIsoCd | mandatory | string | currency ISO code |
| balance | mandatory | long | balance ( in cents per currency)  KRW 100 means KRW 100  USD 100 means USD 1.00 |
| exTransactionId | mandatory | string | external transaction id in the IMS  “0” if transaction id from XTD doesn’t exist in the IMS. |

Example

|  |
| --- |
| {  "returnCode": 0,  "returnMessage": null,  "currencyIsoCd": "THB",  "balance": 5100,  "exTransactionId": "1"  } |

* + 1. CheckTransactionStatus (GET)

1. **Mount Point:** /XTD/Player.svc/transactionId/{transactionId}/Status

This API is to validate the access token.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| transactionId | mandatory | string | transaction id in the XTD |

Example

|  |
| --- |
| /XTD/Player.svc/transactionId/xtd\_transactionId\_7/Status |

***Response***

Response Parameters

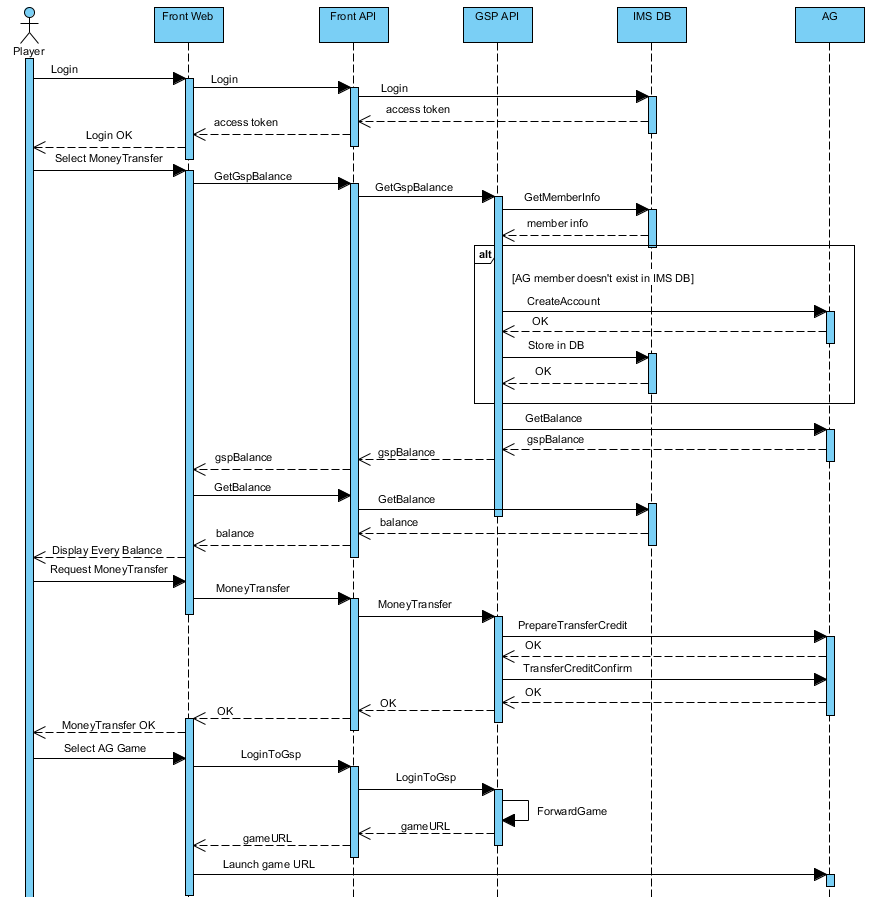
|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| exTransactionId | mandatory | string | external transaction id in the IMS  “0” if transaction id from XTD doesn’t exist in the IMS. |
| isSuccessedTransactionId | mandatory | bool | true: transactionId processed successfully  false: transactionId didn’t executed or error occured |

Example

|  |
| --- |
| {  "exTransactionId": "323534",  "isSuccessedTransactionId": true  } |

* 1. Asia Gaming (AG)

TODO: replace sequence diagram



|  |  |  |  |
| --- | --- | --- | --- |
| **Reference Document** | **Version** | **File Name** | **API** |
| API Document for AG Game Integration | 2.5 | 20131203AG\_API\_Doc\_2.5\_EN (2).pdf |  |

User-agent is sometimes commented.

Configuration

|  |  |
| --- | --- |
| **KEY** | **VALUE** |
| cagent | gsp id |
| MD5\_KEY | 123123 (MerchPwd2 in tb\_game\_list )  Key = MD5(params +” MD5\_Encrypt\_key”);  Attation:  1. md5 is 32 lower case characters.  eg:MD5=cbff0df829bf59470950640f933af9c7  2. ‘MD5\_Encrypt\_key’ value please see the readme.txt doc. |
| DES\_KEY | 12345678 ( MerchPwd3 in tb\_game\_list) |
| host | <http://125.141.206.56:81> ( MerchPwd in tb\_game\_list) |
| User-Agent | WEB\_LIB\_GI\_" +cagent |

* + 1. CreatePlayerTicket (Post) – From IMS to AG

**Mount Point:** http://gi.aggpok.com:81/resource/player-tickets.ucs

The API is for get a player ticket in the AG by calling from Front Web.

If the gspMember exist in the IMS, this API just validate the account and the password.

if not, this API create the account in the AG and get the player ticket.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| username | mandatory | string | player Id |
| serverSessionTicket | mandatory | string | accessToken |
| createPlayer | mandatory | boolean | Create a player if one with the passed in username does not exists. |
| currency | mandatory | string | currency ISO code |

Example

|  |
| --- |
|  |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| id | mandatory | long | the id of the created ticket |
| value | mandatory | string | the ticket’s value |
| player | mandatory | Player |  |

Player

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| id | mandatory | long | the id of the created ticket |
| username | mandatory | string | the username of the player for whom the ticket was created |
| balance | optional | double | the account balance of ther player |

Response Codes

|  |  |  |  |
| --- | --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **Error Code** | **DESCRIPTION** |
| 201 | Created | N/A |  |
| 400 | BAD REQUEST | 1000 | Missing Required parameter |
| 400 | BAD REQUEST | 1016 | the player’s account is inactive |
| 400 | BAD REQUEST | 2002 | the passed in username does not exist on the Live Dealer System |
| 500 | Internal Server Error | 9999 | an error on the server occurred trying to fultill the request |

Example

|  |
| --- |
| Response - example  Status: 201 Created  Headers :  Content-Type: application/xml;charset=UTF-8  Body:  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  <playerTicket>  <id>{long}</id>  <value>{string}</value>  <player>  <id>{long}</id>  <value>{string}</value>  <balance>{double}</balance> - for extra option, check player first balance from your website  <player>  </playerTicket> |

* + 1. ForwardGame (Form POST)

**Mount Point:** Http:// gi.aggpok.com:81/forwardGame.do?params=XXXXXXXXXXXXXXX&key=XXXXXXXXXX

params=des.encrypt(“cagent=TST\_AGIN/\\\\/loginname=test/\\\\/actype=0/\\\\/password=XXXXXX/\\\\/dm=www.xxx.com/\\\\/sid=TST\_AGIN1234567890987/\\\\/gameType=1/\\\\/oddtype=XXX”);

The API is used to load the game page.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| cagent | mandatory | varchar(10) | merch id given from AG |
| loginname | mandatory | varchar(20) | gspMember id  Not underscore |
| password | mandatory | varchar(20) | gspMember password |
| dm | mandatory | string | representative the partner’s website domain  ex) front web site domain |
| sid | mandatory | string | sid = cagent + sequence  sequence = 13~16 digit, unique |
| actype | mandatory | varchar(1) | 0: trial account  1: real account |
| gameType | optional | int | gameType means the game type  value (AGIN Game)  (为空将导入整合页面)enmpty will show all games  0 大厅  1 AGQ 厅  2 DSP 厅  3 自选多台  4 VIP 厅  value(BBIN Game)  (为空将导入整合页面)enmpty will show all games  1: 视讯  2: 彩票  3：BB 彩票  4：球类  5: 电子游艺  (Option 可选) |
| lang | optional | int | Web Browser Language ISO code  default: 1(zh-cn)  h-cn(中文) 1  zh-tw(中文台湾) 2  en-us(英语) 3  euc-jp(日语) 4  ko(韩语) 5  th(泰文) 6  es(西班牙文) 7  vi(越南文) 8  khm(柬埔寨) 9  lao(老挝) 10  id(印尼语) 11  myr(马来西亚) 12  es(西班牙) 13  mx(墨西哥) 14  de(德语) 15  fr(法文) 16  el(希腊文) 17  it(意大利文) 18  pl(波兰文) 19  ru(俄语) 20  hu(匈牙利文) 21  ro(罗马尼亚语) 22 |
| oddtype | optional | string | set the member bet amount range when place bet  Default value: A  A (20~10000)  B (50~5000)  C (100~10000)  D (200~20000)  E (300~30000)  F (400~40000)  G (500~50000)  H (1000~100000)  I (2000~200000)  N (10000~500000) |
| cur | mandatory | string | gspMember’s currency ISO code |

Example

|  |
| --- |
|  |

#### Response

No result, using form post method to submit this URL, you will come into the game page.

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
|  |

* + 1. GetBalance (GET)

**Mount Point:** ~/doBusiness.do?params={EncryptedParams}&key={key}

The API is used to get the member’s available balance in the AG.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| cagent | mandatory | varchar(10) | gsp id |
| loginname | mandatory | varchar(20) | gspMember id |
| method | mandatory | varchar(2) | method representing value: gb |
| actype | mandatory | varchar(1) | money account type  0: trial play account  1: real money account |
| password | mandatory | varchar(20) | gspMember password |
| cur | mandatory | varchar(3) | gspMember’s currency ISO code |

Example

|  |
| --- |
|  |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| info | mandatory | string | Successful, will return the balance, hold 2 digits after the point. |
| msg | mandatory | string | message |

|  |  |  |  |
| --- | --- | --- | --- |
| **INFO VALUE** | **INFO VALUE DESCRIPTION** | **MSG VALUE** | **MSG VALUE DESCRIPTION** |
| 000.00 | Successful,will return the balance  hold 2 digits after the point, e.g. 100.00 | string | error description  If invoke fail, msg show the  fail reason  If invoke succeful, msg is null |
| key\_error | the key value is error |
| account\_not\_exist | The “loginname” which you want to query credit  is not exist |
| network\_error | Network has some problem,lost information |
| error | other error,pls reference “msg” description |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
|  |

* + 1. PostTransfer (GET)

**Mount Point:** ~/doBusiness.do?params={EncryptedParams}&key={key}

The API is used to prepare to transfer credit from IMS to AG, or from AG to IMS.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| cagent | mandatory | varchar(10) | gsp id |
| loginname | mandatory | varchar(20) | gspMember id |
| method | mandatory | varchar(2) | method representing value: tc |
| billno | mandatory | string | billno = cagent + sequence  sequence = 13~16 digit, unique  for BBIN Game: 19 digit  What happen if request with same billno again? |
| type | mandatory | varchar(3) | IN: IMS -> AG  OUT: AG -> IMS |
| credit | mandatory | decimal | the transaction amount  2 decimal point |
| actype | mandatory | varchar(1) | 0: trial account  1: real account |
| password | mandatory | varchar(20) | gspMember password |
| cur | mandatory | varchar(3) | gspMember’s currency ISO code |

Example

|  |
| --- |
|  |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| info | mandatory | string |  |
| msg | mandatory | string | message |

|  |  |  |  |
| --- | --- | --- | --- |
| **INFO VALUE** | **INFO VALUE DESCRIPTION** | **MSG VALUE** | **MSG VALUE DESCRIPTION** |
| 0 | Successful | string | error description  If invoke fail, msg show the  fail reason  If invoke succeful, msg is null |
| key\_error | the key value is error |
| duplicate\_transfer | Duplicate transfer credit |
| account\_not\_exist | The “loginname” which you want to query credit  is not exist |
| network\_error | Network has some problem,lost information |
| not\_enough\_credit | Not enough money to withdraw |
| error | other error,pls reference “msg” description |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
|  |

* 1. UC8 (UC8)
     1. Slot
        1. LaunchGame (POST)

**Mount Point:** API endpoint provided from UC8 slot

The API is for get the game launching url.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameToken | mandatory | string | access token |
| gameCode | mandatory | string | the game that you wishes to launch. |
| langCode | mandatory | string | language code for game  Default: en-us  en\_US: English  zh\_CN: Chinese Simplified  zh\_TW: Chinese Traditional  js\_JP: Japanese  id\_ID: Indonesia  ko\_KR: Korean  th\_TH: Thai |
| currencyCode | mandatory | string | currency ISO code  USD: United State Dollar  RMB: Yuan Renmnibi  KRW: The South Korean Won  THB: Thai Bat  HKD: Hong Kong Dollar |
| partnerId | mandatory | string | operator Id given by UC8 |
| funmode | optional | bool | true: demo gameplay  false: real mode |
| platformId | mandatory | string | 1: PC  2: Mobile / Tablet devices |
| allowSound | optional | bool | false: for staring game silence  true: start game with sounds |

Example

|  |
| --- |
| curl -X POST -d '{  "gameCode":"SlotMachine\_AtlanticTreasures",  "langCode":"us\_EN",  "partnerId":"SPxl",  "funMode":true,  "platformId":2,  "allowSound":false,  "gameToken  ":"534b68be-­‐cd1b-­‐47c9-­‐ac93-­‐7c5d2061b67f"  }' https://api.launchurl.com -H "Content-Type:application/json" |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| url | mandatory | string | full url page that will launch the html game |
| errorCode | mandatory | int | error code |
| errorMessage | mandatory | string | error message |

Example

|  |
| --- |
| {  "url":"http://www.someurl.com/play/AtlanticTreasures?gameCode=SlotMachine\_AtlanticTreasures&langCode=en\_US&token=3e5c08f7-­‐37f0-­‐4d8c-­‐897f-­‐3955e87a4b84&platformId=2",  "errorCode":0,  "errorMessage":null  } |

* + - 1. Authentication (POST)

**Mount Point: ~**/SGS/GetPlayerInfo

The API is for authenticate the token which is passed when a player launch the game.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameToken | mandatory | string | access token |
| requestId | mandatory | string | an unique identifier for each request |
| timestamp | mandatory | string | the exact time the request was send in UTC. |
| sig | mandatory | string | the MD5 for security purpose |
| gameCode | mandatory | string | the current game |
| langCode | mandatory | string | language code for display |

Example

|  |
| --- |
| curl -X POST -d '{  "gameToken":"534b68be-­‐cd1b-­‐47c9-­‐ac93-­‐7c5d2061b67f",  "requestId":4,  "timeStamp":"20150720131025",  "sig":  "c57f85bcef48db0d65f6d2a9d5fa1c62",  "langCode":"en\_US",  "gameCode":"SlotMachine\_AtlanticTreasures"  } ' https://www.merchant.com/SGS/GetPlayerInfo  -­‐H "Content-Type:application/json" |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errorCode | mandatory | int | error code |
| errorMessage | mandatory | string | error message |
| errorAction | optional | string | action code |
| requestId | mandatory | string | the same unique request id that was sent with the request |
| timestamp | mandatory | string | the exact time the response was send in UTC |
| externalId | mandatory | string | player Id |
| name | optional | string | player Id |
| countryCode | mandatory | string | country ISO 3166 alpha-3 |
| balance | mandatory | int | player’s balance in cents |

Example

|  |
| --- |
| {  "gameToken":"534b68be-­‐cd1b-­‐47c9-­‐ac93-­‐7c5d2061b67f",  "requestId":6,  "timeStamp":"20150720131025",  "errorCode":0,  "errorMessage":null,  "errorAction":null,  "externalId":"PL1554",  "CountryCode":"THA",  "name":null,  "balance":152.33,  "currencyCode":"USD"  } |

* + - 1. PlayerBalance (POST)

**Mount Point: ~**/SGS/GetBalance

The API is for retrieve the player’s balance in the IMS system.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameToken | mandatory | string | access token |
| requestId | mandatory | string | an unique identifier for each request |
| timestamp | mandatory | string | the exact time the request was send in UTC. |
| sig | mandatory | string | the MD5 for security purpose |

Example

|  |
| --- |
| curl -X POST -d '{  "gameToken":"534b68be-­‐cd1b-­‐47c9-­‐ac93-­‐7c5d2061b67f",  "requestId":5,  "timeStamp":"20150720131025",  "sig":  "612d6a4fd7c63947b4315d2fc8977ac2"  } ' https://www.merchant.com/SGS/GetBalance -H "Content-  Type:application/json" |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errorCode | mandatory | int | error code |
| errorMessage | mandatory | string | error message |
| errorAction | optional | string | action code |
| requestId | mandatory | string | the same unique request id that was sent with the request |
| timestamp | mandatory | string | the exact time the response was send in UTC |
| balance | mandatory | int | player’s balance in cents |

Example

|  |
| --- |
| {  "gameToken":"534b68be-­‐cd1b-­‐47c9-­‐ac93-­‐7c5d2061b67f",  "requestId":5,  "timeStamp":"20150720131036",  "errorCode":0,  "errorMessage":null,  "errorAction:null,  "balance":200  } |

* + - 1. ProcessBet (POST)

**Mount Point: ~**/SGS/ProcessBet

The API is for authenticate the token which is passed when a player launch the game.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameToken | mandatory | string | access token |
| requestId | mandatory | string | an unique identifier for each request |
| timestamp | mandatory | string | the exact time the request was send in UTC. |
| sig | mandatory | string | the MD5 for security purpose |
| ticketId | mandatory | Int64 | transaction id |
| betAmount | mandatory | int | bet amount in cents |
| winAmount | mandatory | int | win amount in cents |
| transactionType | mandatory | string | InstantWin:  Free\_InstantWin: |
| gameCode | mandatory | string | game id that player places a bet |

Example

|  |
| --- |
| curl -X POST -d '{  "gameToken":"534b68be-­‐cd1b-­‐47c9-­‐ac93-­‐7c5d2061b67f",  "requestId":6,  "timeStamp":"20150720131025",  "sig":  "19f69a6160da1a5843e1cd52aba60356",  "ticketId":175,  "betAmount":0.01,  "winAmount":0.5,  "transactionType":"InstantWin"  }' (ProcessBetUrl) -H "Content-Type:application/json" |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errorCode | mandatory | int | error code |
| errorMessage | mandatory | string | error message |
| errorAction | optional | string | action code |
| requestId | mandatory | string | the same unique request id that was sent with the request |
| timestamp | mandatory | string | the exact time the response was send in UTC |
| extTransactionId | mandatory | string | transaction sequence number in the IMS |
| balance | mandatory | int | player’s balance in cents |

Example

|  |
| --- |
| {  "gameToken":"534b68be-­‐cd1b-­‐47c9-­‐ac93-­‐7c5d2061b67f",  "requestId":6,  "timeStamp":"20150720131025",  "errorCode":0,  "errorMessage":null,  "errorAction":null,  "balance":150.33,  "extTransactionId":"154547454"  } |

* + - 1. ProcessError (POST)

**Mount Point: ~**/SGS/ProcessError

The API is for authenticate the token which is passed when a player launch the game.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameToken | mandatory | string | access token |
| requestId | mandatory | string | an unique identifier for each request |
| timestamp | mandatory | string | the exact time the request was send in UTC. |
| errorCode | mandatory | int | error code |
| errorMessage | mandatory | string | error message |
| requestName | mandatory | string | request API name |

Example

|  |
| --- |
| curl -X POST -d '{  "gameToken":"534b68be-­‐cd1b-­‐47c9-­‐ac93-­‐7c5d2061b67f",  "requestId":4,  "timeStamp":"20150720131025",  "errorCode":175,  "errorMessage":"Invalid Currency",  "requestName":"AuthenticatePlayer"  }'  https://www.merchant.com/SGS/ProcessError -H "  Content-Type:application/json" |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| requestId | mandatory | string | the same unique request id that was sent with the request |

Example

|  |
| --- |
| {  "requestId":4  } |

* + 1. LiveCasino
       1. LaunchGame (POST)

**Mount Point:** lobby URL set in the configuration

The API is for get the game launching URL.

// http://www.vazagaming.com/lobby/game\_gold\_lobby\_live.php?token=82CFB16BD08B48DCB68C6665F7C5E67E&language=EN&operatorID=30115

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| token | mandatory | string | access token |
| language | mandatory | string | language code for game  Default: en-us  EN: English  ZH: Chinese Simplified  CH: Chinese Traditional  js\_JP: Japanese  id\_ID: Indonesia  ko\_KR: Korean  th\_TH: Thai |
| currencyCode | mandatory | string | currency ISO code  USD: United State Dollar  RMB: Yuan Renmnibi  KRW: The South Korean Won  THB: Thai Bat  HKD: Hong Kong Dollar |
| partnerId | mandatory | string | operator Id given by UC8 |
| funmode | optional | bool | true: demo gameplay  false: real mode |
| platformId | mandatory | string | 1: PC  2: Mobile / Tablet devices |
| allowSound | optional | bool | false: for staring game silence  true: start game with sounds |

Example

|  |
| --- |
| curl -X POST -d '{  "gameCode":"SlotMachine\_AtlanticTreasures",  "langCode":"us\_EN",  "partnerId":"SPxl",  "funMode":true,  "platformId":2,  "allowSound":false,  "gameToken  ":"534b68be-­‐cd1b-­‐47c9-­‐ac93-­‐7c5d2061b67f"  }' https://api.launchurl.com -H "Content-Type:application/json" |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| url | mandatory | string | full url page that will launch the html game |
| errorCode | mandatory | int | error code |
| errorMessage | mandatory | string | error message |

Example

|  |
| --- |
| {  "url":"http://www.someurl.com/play/AtlanticTreasures?gameCode=SlotMachine\_AtlanticTreasures&langCode=en\_US&token=3e5c08f7-­‐37f0-­‐4d8c-­‐897f-­‐3955e87a4b84&platformId=2",  "errorCode":0,  "errorMessage":null  } |

* + - 1. Authentication (POST)

**Mount Point: ~**/SGS/GetPlayerInfo

The API is for authenticate the token which is passed when a player launch the game.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameToken | mandatory | string | access token |
| requestId | mandatory | string | an unique identifier for each request |
| timestamp | mandatory | string | the exact time the request was send in UTC. |
| sig | mandatory | string | the MD5 for security purpose |
| gameCode | mandatory | string | the current game |
| langCode | mandatory | string | language code for display |

Example

|  |
| --- |
| curl -X POST -d '{  "gameToken":"534b68be-­‐cd1b-­‐47c9-­‐ac93-­‐7c5d2061b67f",  "requestId":4,  "timeStamp":"20150720131025",  "sig":  "c57f85bcef48db0d65f6d2a9d5fa1c62",  "langCode":"en\_US",  "gameCode":"SlotMachine\_AtlanticTreasures"  } ' https://www.merchant.com/SGS/GetPlayerInfo  -­‐H "Content-Type:application/json" |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errorCode | mandatory | int | error code |
| errorMessage | mandatory | string | error message |
| errorAction | optional | string | action code |
| requestId | mandatory | string | the same unique request id that was sent with the request |
| timestamp | mandatory | string | the exact time the response was send in UTC |
| externalId | mandatory | string | player Id |
| name | optional | string | player Id |
| countryCode | mandatory | string | country ISO 3166 alpha-3 |
| balance | mandatory | int | player’s balance in cents |

Example

|  |
| --- |
| {  "gameToken":"534b68be-­‐cd1b-­‐47c9-­‐ac93-­‐7c5d2061b67f",  "requestId":6,  "timeStamp":"20150720131025",  "errorCode":0,  "errorMessage":null,  "errorAction":null,  "externalId":"PL1554",  "CountryCode":"THA",  "name":null,  "balance":152.33,  "currencyCode":"USD"  } |

* + - 1. PlayerBalance (POST)

**Mount Point: ~**/SGS/GetBalance

The API is for retrieve the player’s balance in the IMS system.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameToken | mandatory | string | access token |
| requestId | mandatory | string | an unique identifier for each request |
| timestamp | mandatory | string | the exact time the request was send in UTC. |
| sig | mandatory | string | the MD5 for security purpose |

Example

|  |
| --- |
| curl -X POST -d '{  "gameToken":"534b68be-­‐cd1b-­‐47c9-­‐ac93-­‐7c5d2061b67f",  "requestId":5,  "timeStamp":"20150720131025",  "sig":  "612d6a4fd7c63947b4315d2fc8977ac2"  } ' https://www.merchant.com/SGS/GetBalance -H "Content-  Type:application/json" |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errorCode | mandatory | int | error code |
| errorMessage | mandatory | string | error message |
| errorAction | optional | string | action code |
| requestId | mandatory | string | the same unique request id that was sent with the request |
| timestamp | mandatory | string | the exact time the response was send in UTC |
| balance | mandatory | int | player’s balance in cents |

Example

|  |
| --- |
| {  "gameToken":"534b68be-­‐cd1b-­‐47c9-­‐ac93-­‐7c5d2061b67f",  "requestId":5,  "timeStamp":"20150720131036",  "errorCode":0,  "errorMessage":null,  "errorAction:null,  "balance":200  } |

* + - 1. ProcessBet (POST)

**Mount Point: ~**/SGS/ProcessBet

The API is for authenticate the token which is passed when a player launch the game.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameToken | mandatory | string | access token |
| requestId | mandatory | string | an unique identifier for each request |
| timestamp | mandatory | string | the exact time the request was send in UTC. |
| sig | mandatory | string | the MD5 for security purpose |
| ticketId | mandatory | Int64 | transaction id |
| betAmount | mandatory | int | bet amount in cents |
| winAmount | mandatory | int | win amount in cents |
| transactionType | mandatory | string | InstantWin:  Free\_InstantWin: |
| gameCode | mandatory | string | game id that player places a bet |

Example

|  |
| --- |
| curl -X POST -d '{  "gameToken":"534b68be-­‐cd1b-­‐47c9-­‐ac93-­‐7c5d2061b67f",  "requestId":6,  "timeStamp":"20150720131025",  "sig":  "19f69a6160da1a5843e1cd52aba60356",  "ticketId":175,  "betAmount":0.01,  "winAmount":0.5,  "transactionType":"InstantWin"  }' (ProcessBetUrl) -H "Content-Type:application/json" |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errorCode | mandatory | int | error code |
| errorMessage | mandatory | string | error message |
| errorAction | optional | string | action code |
| requestId | mandatory | string | the same unique request id that was sent with the request |
| timestamp | mandatory | string | the exact time the response was send in UTC |
| extTransactionId | mandatory | string | transaction sequence number in the IMS |
| balance | mandatory | int | player’s balance in cents |

Example

|  |
| --- |
| {  "gameToken":"534b68be-­‐cd1b-­‐47c9-­‐ac93-­‐7c5d2061b67f",  "requestId":6,  "timeStamp":"20150720131025",  "errorCode":0,  "errorMessage":null,  "errorAction":null,  "balance":150.33,  "extTransactionId":"154547454"  } |

* + - 1. ProcessError (POST)

**Mount Point: ~**/SGS/ProcessError

The API is for authenticate the token which is passed when a player launch the game.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| gameToken | mandatory | string | access token |
| requestId | mandatory | string | an unique identifier for each request |
| timestamp | mandatory | string | the exact time the request was send in UTC. |
| errorCode | mandatory | int | error code |
| errorMessage | mandatory | string | error message |
| requestName | mandatory | string | request API name |

Example

|  |
| --- |
| curl -X POST -d '{  "gameToken":"534b68be-­‐cd1b-­‐47c9-­‐ac93-­‐7c5d2061b67f",  "requestId":4,  "timeStamp":"20150720131025",  "errorCode":175,  "errorMessage":"Invalid Currency",  "requestName":"AuthenticatePlayer"  }'  https://www.merchant.com/SGS/ProcessError -H "  Content-Type:application/json" |

***Response***

Response Parameters

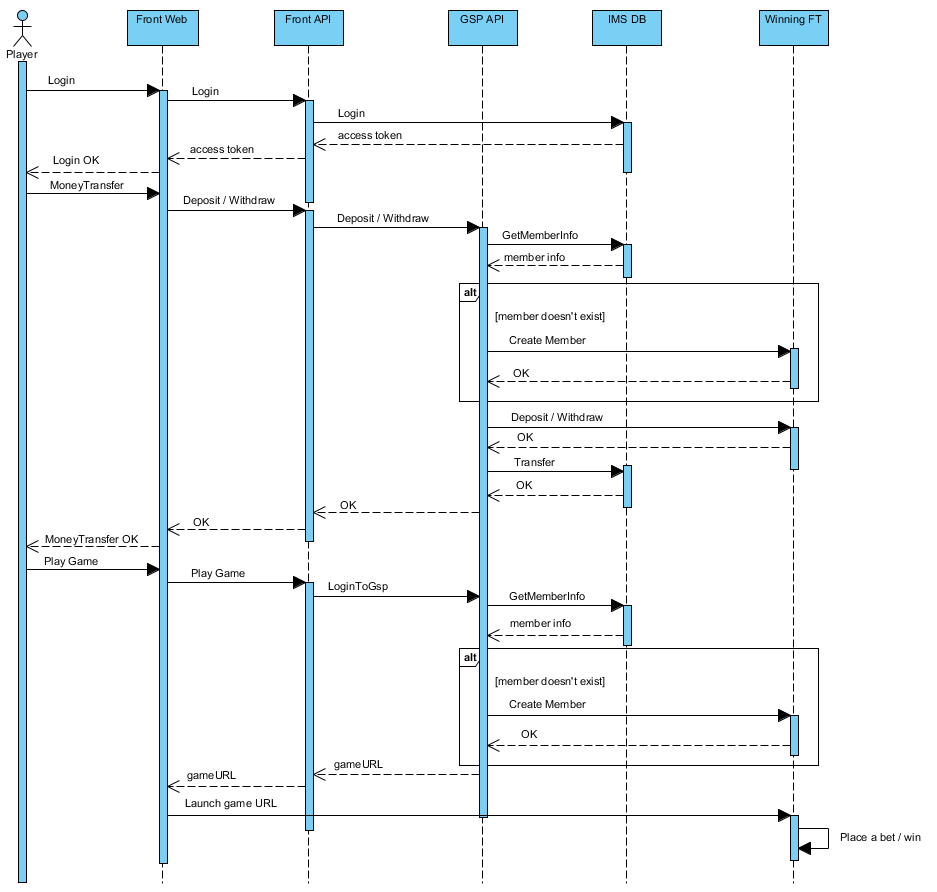
|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| requestId | mandatory | string | the same unique request id that was sent with the request |

Example

|  |
| --- |
| {  "requestId":4  } |

# **WalletToWallet API**

* 1. Winning FT (WFT)



|  |  |  |  |
| --- | --- | --- | --- |
| **Reference Document** | **Version** | **File Name** | **API** |
| Sportsbook Interation Methods |  | Sportsbook Interation Methods.doc |  |
| WFT API (2012-01-07) |  | WFT API - 2013-04-29.pdf |  |

General Error Codes

|  |  |
| --- | --- |
| **Error ID** | **Error Description** |
| -1 | System maintenance |
| -2 | Invalid secret |
| -3 | Ip address not allow |
| -4 | Invalid username (length 1-14, format A-Za-z0-9)"; |
| -5 | Invalid action |
| -7 | Invalid agent |
| -8 | Invalid username"; |
| -100 | Invalid Intput |
| -101 | Exception |

**connectionURL: merchPassword2 of tb\_game\_list**

* + 1. CreateMember (GET)

**Mount Point:** {**connectionURL**}?agent={agent}&secret={secret}&action=create&username={username}&currency={currency}

The API is used to create new member account on Winning Sports Game Service Provider.

#### Request

Request Parameters

**connectionURL: merchPassword2 of tb\_game\_list**

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | p@ar |
| secret | mandatory | string | su84ruXV |
| action | mandatory | string | create |
| username | mandatory | string | gspMember id |
| currency | mandatory | string | currency ISO code |

Example

|  |
| --- |
| http://hapi.bm.1sgames.com/api.aspx??agent=p%40ar&secret=su84ruXV&action=create&username=test100&currency=USD |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  1: Username exist  2: Invalid currency  + general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | gspMember id |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>-7</errcode>  <errtext>Invalid agent (empty)</errtext>  <result></result>  </response> |

* + 1. Update (GET)

**Mount Point**:

{**connectionURL**}?agent={agent}&secret={secret}&action=**update**&username={username}&max1={max1}& max2={max2}&max3={max3}&max4={max4}&lim1={lim1}&lim2={lim2}&lim3={lim3}&comtype ={comtype}&com1={com1}&com2={com2}&com3={com3}&suspend={suspend}

The API is used to update a member’s max bets, limits and commission settings.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | p@ar |
| secret | mandatory | string | su84ruXV |
| action | mandatory | string | update |
| username | mandatory | string | gspMember id |
| max1 | mandatory | string | Max bet for HDP/OU/OE |
| max2 | mandatory | string | Max bet for 1X2 |
| max3 | mandatory | string | Max bet for PAR |
| max4 | mandatory | string | Max bet for ORT/CS/TG/HFT/FLG |
| lim1 | mandatory | string | Per match limit for all except PAR/ORT |
| lim2 | mandatory | string | Per match payout limit for PAR (payout = bet \* odds) |
| lim3 | mandatory | string | Per match limit for ORT |
| lim4 | mandatory | string |  |
| comtype | mandatory | string | Choice of A,B,C,D,E,F,4 for HDP/OU/OE |
| com1 | mandatory | string | Commission for HDP/OU/OE |
| com2 | mandatory | string | Commisson for 1X2/ORT |
| com3 | mandatory | string | Commission for PAR/CS/TG/HFT/FLG |
| suspend | mandatory | string | 0: no suspend, 1:suspend |

Example

|  |
| --- |
| http://hapi.bm.1sgames.com/api.aspx?agent=p%40ar&secret=su84ruXV&action=update&username=test100&max1=1100&%20max2=1200&max3=1300&max4=1400&lim1=1000&lim2=1100&lim3=1200&comtype=A&com1=0.1&com2=0.2&com3=0.3&suspend=0 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error |
| errtext | mandatory | string | error message |
| result | mandatory | string | gspMember id |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>-100</errcode>  <errtext>Empty max2</errtext>  <result></result>  </response> |

* + 1. CheckBalance (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=balance&username={username}&currency={currency}

The API is used to look up current balance of a member.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | p@ar |
| secret | mandatory | string | su84ruXV |
| action | mandatory | string | balance |
| username | mandatory | string | gspMember id |
| currency | mandatory | string | gspMember currency ISO code |

Example

|  |
| --- |
| http://hapi.bm.1sgames.com/api.aspx?agent=p%40ar&secret=su84ruXV&action=balance&username=test100 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  1: User not exist  2: Account locked  3: System maintainenace  4: Egame online  + general error |
| errtext | mandatory | string | error message |
| result | mandatory | string | balance |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>1</errcode>  <errtext>User not exist</errtext>  <result>0</result>  </response> |

* + 1. Deposit

**Mount Point:**

{**connectionURL**}?agent={agent}&secret={secret}&action=deposit&username={username}&amount={amount}&serial={ serial}

The API is used to perform fund in.

You must call "Check Balance" with no error before deposit.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | p@ar |
| secret | mandatory | string | su84ruXV |
| action | mandatory | string | deposit |
| username | mandatory | string | gspMember id |
| amount | mandatory | string | transaction amount |
| serial | mandatory | string | transaction id |

Example

|  |
| --- |
| http://hapi.bm.1sgames.com/api.aspx?agent=p%40ar&secret=su84ruXV&action=deposit&username=test100&amount=100&serial=100 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code |
| errtext | mandatory | string | error message |
| result | mandatory | string | result of API |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>1</errcode>  <errtext>Cannot insert the value NULL into column 'AccId', table 'WSPJas.dbo.Payments'; column does not allow nulls. INSERT fails.  The statement has been terminated.</errtext>  <result>0</result>  </response> |

* + 1. Withdraw (GET)

**Mount Point:** {**connectionURL**}?agent={agent}&secret={secret}&action=withdraw&username={username}&amount={amount}&serial={serial}

The API is used to perform fund out.

You must call "Check Balance" with no error before deposit.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | p@ar |
| secret | mandatory | string | su84ruXV |
| action | mandatory | string | withdraw |
| username | mandatory | string | gspMember id |
| amount | mandatory | string | transaction amount |
| serial | mandatory | string | transaction id |

Example

|  |
| --- |
| http://hapi.bm.1sgames.com/api.aspx?agent=p%40ar&secret=su84ruXV&action=withdraw&username=test100&amount=100&serial=100 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code |
| errtext | mandatory | string | error message |
| result | mandatory | string | result of API |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>1</errcode>  <errtext>Cannot insert the value NULL into column 'AccId', table 'WSPJas.dbo.Payments'; column does not allow nulls. INSERT fails.  The statement has been terminated.</errtext>  <result>0</result>  </response> |

* + 1. Login (GET)

**Mount Point**:

{**connectionURL**}?agent={agent}&secret={secret}&action=login&username={**username**}&lang=EN-US&host={**host**}

The API is used to login the game

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | p@ar |
| secret | mandatory | string | su84ruXV |
| action | mandatory | string | login |
| username | mandatory | string | gspMember id |
| lang | mandatory | string | Web Browser Language Identification Codes  EN-US(hard-code) |
| host | mandatory | string | game url |

Example

|  |
| --- |
| http://hapi.bm.1sgames.com/api.aspx?agent=p%40ar&secret=su84ruXV&action=deposit&username=test100&host=sport.eg.1sgames.com&lang=EN-US |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  1: Internal error: invalid LoginPrivateKey |
| errtext | mandatory | string | error message |
| result | mandatory | string | User login url |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>-100</errcode>  <errtext>Empty serial</errtext>  <result></result>  </response> |

* + 1. Logout (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=logout&username={**username**}

The API is used to logout the game.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | p@ar |
| secret | mandatory | string | su84ruXV |
| action | mandatory | string | logout |
| username | mandatory | string | gspMember id |

Example

|  |
| --- |
| http://hapi.bm.1sgames.com/api.aspx?agent=p%40ar&secret=su84ruXV&action=logout&username=test100 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : no error |
| errtext | mandatory | string | error message |
| result | mandatory | string |  |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result></result>  </response> |

* + 1. CheckPayment (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=check\_payment&username={**username**}&serial ={**serial**}

The API is used to check gspMember’s payment.

If username is empty, check all users.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | p@ar |
| secret | mandatory | string | su84ruXV |
| action | mandatory | string | check\_payment |
| username | mandatory | string | gspMember id |
| serial | optional | string | serial |

Example

|  |
| --- |
| http://hapi.bm.1sgames.com/api.aspx?agent=p%40ar&secret=su84ruXV&action=check\_payment&username=test100&serial=001 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | List<Payment> |

Payment

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Username | mandatory | string | gspMember id |
| Amount | mandatory | string | ?? |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>1</errcode>  <errtext>No result</errtext>  <result></result>  </response> |

* + 1. CheckOnlineUsers (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=check\_online&username={**username**}

The API is used to check online members.

If username is empty, check all users.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | p@ar |
| secret | mandatory | string | su84ruXV |
| action | mandatory | string | check\_online |
| username | mandatory | string | gspMember id |

Example

|  |
| --- |
| http://hapi.bm.1sgames.com/api.aspx?agent=p%40ar&secret=su84ruXV&action=check\_online&username=&serial=001 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | List<string>: list of username (gspMember id) |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result></result>  </response> |

* + 1. Ticket (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=ticket&username={**username**}&start ={**start**}& duration={**duration**}&match\_over={**match\_over**}

The API is used to ???

Date and time range is based ticket transaction date.

Maximum 100 tickets can be returned.

If username is empty, check all users.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | p@ar |
| secret | mandatory | string | su84ruXV |
| action | mandatory | string | ticket |
| username | mandatory | string | gspMember id |
| start | mandatory | string | start date time  datetime format?? |
| duration | mandatory | string | number of seconds |
| match\_over | mandatory | string | 0 or 1 |

Example

|  |
| --- |
| http://hapi.bm.1sgames.com/api.aspx?agent=p%40ar&secret=su84ruXV&action=ticket&username=test100&start=2015-08-18+09%3A10%3A00&duration=600&match\_over=0 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | List<Ticket> |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>-101</errcode>  <errtext>This SP not available.</errtext>  <result></result>  </response> |

* + 1. Parlay (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=parlay&username={**username**}&ticket\_id ={**ticket\_id**}

The API is used to ???

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | p@ar |
| secret | mandatory | string | su84ruXV |
| action | mandatory | string | parlay |
| username | mandatory | string | gspMember id |
| ticket\_id | mandatory | string | ticket\_id |

Example

|  |
| --- |
| http://hapi.bm.1sgames.com/api.aspx?agent=p%40ar&secret=su84ruXV&action=parlay&username=test100&ticket\_id=001 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | List<Ticket> |

Example

|  |  |
| --- | --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result></result>  </response> |  |

* + 1. Team (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=team&username={**username**}&team\_id={**team\_id**}

The API is used to get an information of searching team.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | p@ar |
| secret | mandatory | string | su84ruXV |
| action | mandatory | string | team |
| username | mandatory | string | gspMember id |
| team\_id | mandatory | string | team\_id |

Example

|  |
| --- |
| http://hapi.bm.1sgames.com/api.aspx?agent=p%40ar&secret=su84ruXV&action=team&username=test100&team\_id=aaa |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | <name><lang>en-US</lang><txt>Arsenal</txt></name>... |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>-100</errcode>  <errtext>Invalid team\_id: aaa</errtext>  <result></result>  </response> |

* + 1. League (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=league&username={**username**}&league\_id ={**league\_id**}

The API is used to ???

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | p@ar |
| secret | mandatory | string | su84ruXV |
| action | mandatory | string | league |
| username | mandatory | string | gspMember id |
| league\_id | mandatory | string | league\_id |

Example

|  |
| --- |
| http://hapi.bm.1sgames.com/api.aspx?agent=p%40ar&secret=su84ruXV&action=league&username=test100&league\_id=uefa |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | <name><lang>en-US</lang><txt>UEFA</txt></name>... |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>-100</errcode>  <errtext>Invalid league\_id: uefa</errtext>  <result></result>  </response> |

* + 1. Fetch (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=fetch&username={**username**}

The API is used to ???

Maximum 100 tickets can be returned.

Tickets will be available for fetch when ticket is :

(a) Accepted (b) Rejected (c) Cancelled (d) Match over.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | p@ar |
| secret | mandatory | string | su84ruXV |
| action | mandatory | string | fetch |
| username | mandatory | string | gspMember id |

Example

|  |
| --- |
| http://hapi.bm.1sgames.com/api.aspx?agent=p%40ar&secret=su84ruXV&action=fetch&username=test100 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | List<Ticket> |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result>  <ticket>  <fid>37860115</fid>  <id>OU212188527014</id>  <t>2015-08-18 08:58:17.980</t>  <u>p@arCO7819</u>  <b>100</b>  <w>0</w>  <a>0</a>  <c>0</c>  <ip>123.140.40.2</ip>  <league>5750</league>  <home>453530</home>  <away>127864</away>  <status>A</status>  <game>OU</game>  <odds>0.89</odds>  <side>1</side>  <info>1.5</info>  <half>0</half>  <trandate>2015-08-18 08:58:17</trandate>  <workdate>2015-08-17 00:00:00</workdate>  <matchdate>18/08 07:30</matchdate>  <runscore>0-1</runscore>  <score></score>  <htscore></htscore>  <flg></flg>  <res>P</res>  <edesc></edesc>  <eres></eres>  <exrate>0.100000001490116</exrate>  <jp>0</jp>  <oddstype>MY</oddstype>  <sportstype>120</sportstype>  </ticket>  <ticket>  <fid>37860865</fid>  <id>OU212188527014</id>  <t>2015-08-18 09:31:11.040</t>  <u>p@arCO7819</u>  <b>100</b>  <w>-100</w>  <a>100</a>  <c>0</c>  <ip>123.140.40.2</ip>  <league>5750</league>  <home>453530</home>  <away>127864</away>  <status>A</status>  <game>OU</game>  <odds>0.89</odds>  <side>1</side>  <info>1.5</info>  <half>0</half>  <trandate>2015-08-18 08:58:17</trandate>  <workdate>2015-08-17 00:00:00</workdate>  <matchdate>18/08 07:30</matchdate>  <runscore>0-1</runscore>  <score>0-1</score>  <htscore></htscore>  <flg></flg>  <res>LA</res>  <edesc></edesc>  <eres></eres>  <exrate>0.100000001490116</exrate>  <jp>0</jp>  <oddstype>MY</oddstype>  <sportstype>120</sportstype>  </ticket>  </result>  </response> |

* + 1. Mark Fetched (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=mark\_fetched&username={**username**}& fetch\_ids={**fetch\_ids**}

The API is used to ???

Mark tickets as fetched so that it will not be fetched again.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | p@ar |
| secret | mandatory | string | su84ruXV |
| action | mandatory | string | mark\_fetched |
| username | mandatory | string | gspMember id |
| fetch\_ids | mandatory | string | fetch\_ids |

Example

|  |
| --- |
| http://hapi.bm.1sgames.com/api.aspx?agent=p%40ar&secret=su84ruXV&action=mark\_fetched&username=test100&fetch\_ids=37860115 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | none |

Example

|  |  |
| --- | --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result></result>  </response> |  |

* + 1. GetJackpots (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=get\_jackpots&username={**username**}&league\_id ={**league\_id**}

The API is used to get jackpot values.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | p@ar |
| secret | mandatory | string | su84ruXV |
| action | mandatory | string | get\_jackpots |
| username | mandatory | string | gspMember id |

Example

|  |
| --- |
| http://hapi.bm.1sgames.com/api.aspx?agent=p%40ar&secret=su84ruXV&action=get\_jackpots&username=test100&league\_id=uefa |

#### Response

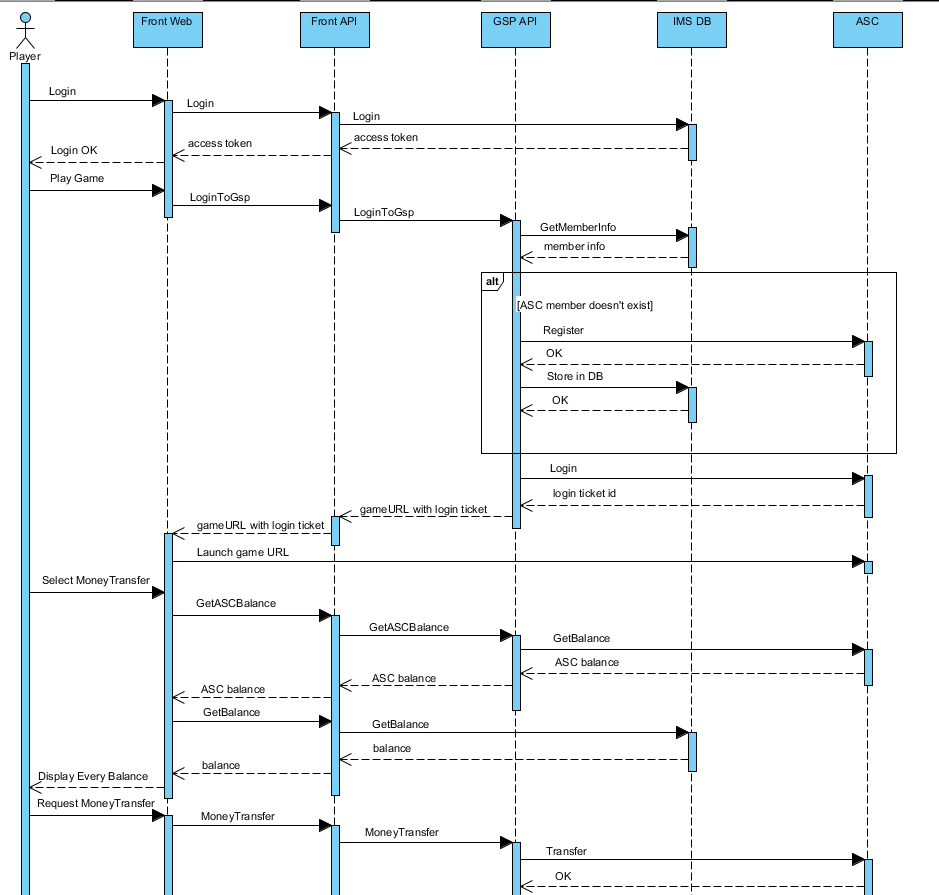
Response Parameters

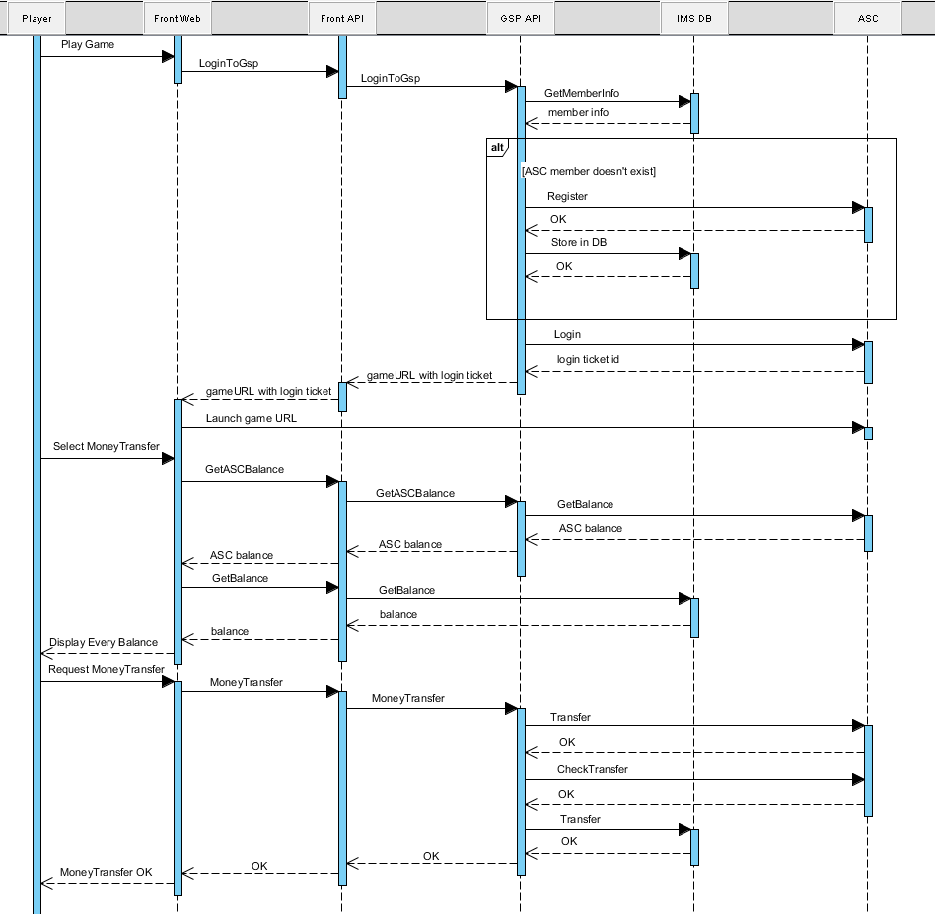
|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | String of jackpot values separated by comma. |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result>6240218,683208,7106247,11746793,41753096,4270499,2317303,116182897,3820360,59039303,47055391,99532661,79551867</result>  </response> |

* 1. Asia Sports Club (ASC)





|  |  |  |  |
| --- | --- | --- | --- |
| **Reference Document** | **Version** | **File Name** | **API** |
| SPORT API REFERENCE | v3 | SportService api\_en\_v3 (1).docx |  |

* + 1. Register

**Mount Point:** ~/**Register**?APIPassword={**password**}&AgentAccount={**test**}&MemberAccount={**test**}&NickName={**test**}&Currency={**RMB**}

The API is used to create a member.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | varchar(32) | api password |
| agentAccount | mandatory | varchar(20) | Gsp id  If you input one account, the system will add the member under this AgentAccount.  If you input nothing, the system will auto find AengtAccount for the member by currency. |
| memberAccount | mandatory | varchar(20) | gspMember id |
| nickName | mandatory | varchar(32) | show name in the sport page |
| currency | mandatory | varchar(10) | currency ISO code |

Example

|  |
| --- |
| http://api.info99bet.com/SportAPI.asmx/Register?APIPassword=f0a3ae129d0685a3073c7c6f5bcc6107&AgentAccount=7pay&MemberAccount=test&NickName=test&Currency=RMB |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | int | error code |
| errtext | mandatory | string | error description |
| result | mandatory | string | result of invoking API |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <response>  <errcode>000003</errcode>  <errtext></errtext>  <result></result>  </response> |

* + 1. RegisterByLimit

**Mount Point:** ~/ RegisterByLimit?APIPassword={**password**}&AgentAccount={**test**}&MemberAccount={**test**}&NickName={**test**}&Currency={**currency**}&MinBet={**minBet**}&MaxBet={**maxBet**}&PerMaxBet={**perMaxBet**}&MixMinBet={**mixMinBet**}&MixMaxBet={**minMaxBet**}&DayLimit={**dayLimit**}&GroupComm={**groupComm**}

The API is used to create a member.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | varchar(32) | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| agentAccount | mandatory | varchar(20) | Gsp id  please check the start part of document.  If you input one account, the system will add the member under this AgentAccount.  If you input nothing, the system will auto find AengtAccount for the member by currency. |
| memberAccount | mandatory | varchar(20) | gspMember id |
| nickName | mandatory | varchar(32) | gspMember id  hard-code: 11  show name in the sport page |
| currency | mandatory | varchar(10) | currency ISO code |
| minBet | mandatory | int | what for?  Will Modify when value>=0;  Not Modify when value-1; |
| maxBet | mandatory | int | what for?  Will Modify when value>=0;  Not Modify when value-1; |
| perMaxBet | int | int | per? each betting max amount  Will Modify when value>=0;  Not Modify when value-1; |
| mixMinBet | int | int | mix?  Will Modify when value>=0;  Not Modify when value-1; |
| mixMaxBet | int | int | mix?  Will Modify when value>=0;  Not Modify when value-1; |
| dayLimit | int | int | dayLimit?  Will Modify when value>=0;  Not Modify when value-1; |
| groupComm | mandatory | varchar(1) | commission group? |

Example

|  |
| --- |
| <http://domain/RegisterByLimit?APIPassword=f0a3ae129d0685a3073c7c6f5bcc6107&&AgentAccount=test>  &MemberAccount=test&NickName=test&Currency=RMB&MinBet=-1&MaxBet=-1&PerMaxBet=-1  &MixMinBet=1&MixMaxBet=10000&DayLimit=-1&GroupComm=A |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | int | error code |
| errtext | mandatory | string | error description |
| result | mandatory | string | result of invoking API |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion=”1.0”encoding=”utf-8”?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result></result>  </response> |

* + 1. Login

**Mount Point:** ~/Login?APIPassword={**apiPassword**}&AgentAccount={**test**}&MemberAccount={**test**}&NickName={**test**}&Currency={**RMB**}

The API is used to login.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | varchar(32) | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| memberAccount | mandatory | varchar(20) | gspMember id |

Example

|  |
| --- |
| <http://domain/sportapi.asmx/Login?APIPassword=f0a3ae129d0685a3073c7c6f5bcc6107>  &MemberAccount=test |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | int | error code |
| errtext | mandatory | string | error description |
| result | mandatory | string | member login ticket id |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion="1.0"encoding="utf-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result>B1D11272E2A915849628515BA045B7C5</result>  </response> |

* + 1. Login Member Site

**Mount Point: ~/LoginCode?** [**http://memberdomain/Memberlogin.aspx?LoginCode=B1D11272E2A915849628515BA045B7C5&Language=ch&PageStyle=1&OddsStyle=1**](http://memberdomain/Memberlogin.aspx?LoginCode=B1D11272E2A915849628515BA045B7C5&Language=ch&PageStyle=1&OddsStyle=1)

The API is used to check the transaction status.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| loginCode | mandatory | varchar(32) | result of 5.2.3 Login |
| language | mandatory | varchar(10) |  |
| pageStyle | mandatory | int | 1: ASC |
| oddsStyle | mandatory | int | odds style  1=my  2=eu  3=hk  4=in |

Example

|  |
| --- |
| <http://memberdomain/Memberlogin.aspx?LoginCode=B1D11272E2A915849628515BA045B7C5>  &Language=ch&PageStyle=1&OddsStyle=1 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code |
| errtext | mandatory | string | error description |
| result | mandatory | varchar(32) | API result |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion=”1.0”encoding=”utf-8”?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result></result>  </response> |

* + 1. Logout

**Mount Point:** ~/Logout?APIPassword={**apiPassword**}&MemberAccount={**test**}

The API is used to logout.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | varchar(32) | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| memberAccount | mandatory | varchar(20) | gspMember id |

Example

|  |
| --- |
| <http://domain/sportapi.asmx/Logout?APIPassword=f0a3ae129d0685a3073c7c6f5bcc6107>  &MemberAccount=test |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | int | error code |
| errtext | mandatory | string | error description |
| result | mandatory | string | member login ticket id |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion="1.0"encoding="utf-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result></result>  </response> |

* + 1. GetOnlineMember

**Mount Point:** ~/GetOnlineMember?APIPassword={**apiPassword**}&MemberAccount={**memberAccount**}

The API is used to get a list online members.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | varchar(32) | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| memberAccount | mandatory | varchar(20) | If you input one account, will return this account is online;  If you input nothing, will return all online members. |

Example

|  |
| --- |
| <http://domain/sportapi.asmx/GetOnlineMember?APIPassword=f0a3ae129d0685a3073c7c6f5bcc6107>  &MemberAccount=test |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | int | error code |
| errtext | mandatory | string | error description |
| result | mandatory | List<Account> | account list |

Account

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Account | mandatory | varchar(32) | account |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion="1.0"encoding="utf-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result>  <Account>account1</Account>  <Account>account2</Account>  </result>  </response> |

* + 1. Transfer

**Mount Point:** ~/Transfer?APIPassword={**apiPassword**}&MemberAccount={**memberAccount**}&SerialNumber={**serialNumber**}

&Amount={**amount**}&TransferType={**transferType**}&Key={**key**}

Note: Receipt of returned result, regardless of success or failure, please check the status of the transfer.

The API is used to transfer money between ASC and IMS.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | varchar(32) | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| memberAccount | mandatory | varchar(20) | gspMember id |
| serialNumber | mandatory | varchar(100) | transaction id from IMS |
| amount | mandatory | decimal | transfer amount of money  Amount must are 4 numbers after dot. |
| transferType | mandatory | int | 0: transfer to ASC (Deposit)  1: transfer from ASC (Withdraw) |
| key | mandatory | varchar(6) | valid code  Last 6 chars of MD5(APIPassword+MemberAccount+Amount)  Amount must are 4 numbers after dot.  e.g:f0a3ae129d0685a3073c7c6f5bcc6107test10000.0000 |

Example

|  |
| --- |
| <http://wwwt.star996.com/sportapi/sportapi.asmx/Transfer?APIPassword>=  f0a3ae129d0685a3073c7c6f5bcc6107&MemberAccount=test&SerialNumber=test00001  &Amount=10000&TransferType=0&Key=9eebc4 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | int | error code |
| errtext | mandatory | string | error description |
| result | mandatory | result | below |

result

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Account | mandatory | varchar(32) | gspMember id |
| SerialNumber | mandatory | varchar(100) | transfer id from ASC |
| Amount | mandatory | decimal | transfer amount |
| Balance | mandatory | decimal | gspMember balance in ASC |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion="1.0"encoding="utf-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result>  <Account>testxtan</Account>  <SerialNumber>test00001</SerialNumber>  <Amount>10000</Amount>  <Balance>10000.0000</Balance>  </result>  </response> |

* + 1. CheckTransfer

**Mount Point:** ~/CheckTransfer?APIPassword={**apiPassword**}&SerialNumber={**serialNumber**}

The API is used to check whether the transfer executed successfully or not.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | varchar(32) | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| serialNumber | mandatory | varchar(100) | transaction id to check the processed result |

Example

|  |
| --- |
| <http://wwwt.star996.com/sportapi/sportapi.asmx/CheckTransfer?APIPassword=f0a3ae129d0685a3073c7>  c6f5bcc6107&SerialNumber=test00001 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | int | error code |
| errtext | mandatory | string | error description |
| result | mandatory | result | below |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion="1.0"encoding="utf-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result></result>  </response> |

* + 1. CheckTransferByAccount

**Mount Point:** ~/CheckTransfer?APIPassword={**apiPassword**}&MemberAccount={**memberAccount**}&SerialNumber={**serialNumber**}

The API is used to check whether the transfer executed successfully or not.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | varchar(32) | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| memberAccount | mandatory | varchar(20) | gspMember id |
| serialNumber | mandatory | varchar(100) | transaction id to check the processed result |

Example

|  |
| --- |
| <http://wwwt.star996.com/sportapi/sportapi.asmx/CheckTransfer?APIPassword=f0a3ae129d0685a3073c7c6>  f5bcc6107&MemberAccount=test&SerialNumber=test00001 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | int | error code |
| errtext | mandatory | string | error description |
| result | mandatory | result | below |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion="1.0"encoding="utf-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result></result>  </response> |

* + 1. GetBalance

**Mount Point:** ~/GetBalance?APIPassword={**apiPassword**}&MemberAccount={**memberAccount**}

The API is used to look up a member’s balance.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | varchar(32) | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| memberAccount | mandatory | varchar(20) | gspMember id |

Example

|  |
| --- |
| <http://domain/sportapi.asmx/GetBalance?APIPassword=f0a3ae129d0685a3073c7c6f5bcc6107>  &MemberAccount=test |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | int | error code |
| errtext | mandatory | string | error description |
| result | mandatory | string | API result |

result

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Account | mandatory | varchar(32) | gspMember id |
| Balance | mandatory | decimal | gspMember balance in ASC |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion=”1.0”encoding=”utf-8”?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result>  <Account>test</Account>  <Balance>0.0000</Balance>  </result>  </response> |

* + 1. UpdateMemberLimit

**Mount Point** ~/UpdateMemberLimit?APIPassword={**apiPassword**}&MemberAccount={**memberAccount**}&GameID ={**gameId**}&MinBet={**minBet**}&MaxBet={**maxBet**}&PerMaxBet={**perMaxBet**}&MixMinBet={**mixMinBet**}&MixMaxBet={**mixMaxBet**}&DayLimit={**dayLimit**}

The API is used to update gspMember’s bet limit.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | string | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| memberAccount | mandatory | string | gspMember id |
| gameId | mandatory | int | game id |
| minBet | mandatory | decimal | what for?  Will Modify when value>=0;  Not Modify when value-1; |
| maxBet | mandatory | decimal | what for?  Will Modify when value>=0;  Not Modify when value-1; |
| perMaxBet | mandatory | decimal | what for?  Will Modify when value>=0;  Not Modify when value-1; |
| mixMinBet | mandatory | decimal | what for?  Will Modify when value>=0;  Not Modify when value-1; |
| mixMaxBet | mandatory | decimal | what for?  Will Modify when value>=0;  Not Modify when value-1; |
| dayLimit | mandatory | decimal | what for?  Will Modify when value>=0;  Not Modify when value-1; |

Example

|  |
| --- |
| http://domain/sportapi.asmx/UpdateMemberLimit?APIPassword=f0a3ae129d0685a3073c7c6f5bcc6107&MemberAccount=test&MinBet=-1&MaxBet=-1&PerMaxBet=-1&MixMinBet=1&MixMaxBet=10000&DayLimit=-1 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code |
| errtext | mandatory | string | error description |
| result | mandatory | varchar(32) | API result |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion="1.0"encoding="utf-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result></result>  </response> |

* + 1. GetMemberLimit

**Mount Point** ~/GetMemberLimit?APIPassword={**apiPassword**}&MemberAccount={**memberAccount}**

The API is used to get gspMember’s bet limit information.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | varchar(32) | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| memberAccount | mandatory | string | gspMember id |

Example

|  |
| --- |
| <http://domain/sportapi.asmx/GetBalance?APIPassword=f0a3ae129d0685a3073c7c6f5bcc6107>  &MemberAccount=test |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code |
| errtext | mandatory | string | error description |
| result | mandatory |  | API result |

result

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| MinBet | mandatory | int | minimum bet |
| MaxBet | mandatory | int | maximum bet |
| PerMaxBet | mandatory | int | per maximum bet |
| MixMinBet | mandatory | int | mix minimum bet |
| MixMaxBet | mandatory | int | mix maximum bet |
| DayLimit | mandatory | int | day limit |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion="1.0"encoding="utf-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result>  <MinBet>5.0000</MinBet>  <MaxBet>555.0000</MaxBet>  <PerMaxBet>5555.0000</PerMaxBet>  <MixMinBet>0</MixMinBet>  <MixMaxBet>0</MixMaxBet>  <DayLimit>0</DayLimit>  </result>  </response> |

* + 1. GetBetSheet

**Mount Point** ~/GetBetSheet?APIPassword={**apiPassword**}&MemberAccount={**memberAccount}&**BetId={**betId**}&Status={**status**}& FromDate={**fromDate**}&ToDate={**toDate**}

The API is used to get gspMember’s bet limit information.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | varchar(32) | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| memberAccount | mandatory | string | gspMember id |
| betId | mandatory | long | bet no |
| status | mandatory | int | 0:ALL  1:Running  2:Settled  3:Rejected |
| fromDate | mandatory | datetime | yyyy-MM-dd HH:mm:ss |
| toDate | mandatory | datetime | yyyy-MM-dd HH:mm:ss  FromDate-ToDate<=30mins |

Example

|  |
| --- |
| http://domain/sportapi.asmx/GetBetSheet?APIPassword=f0a3ae129d0685a3073c7c6f5bcc6107&MemberAccount=&BetID=0&Status=-1&FromDate=2013-08-27 10:0:0&ToDate=2013-08-27 10:30:0 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code |
| errtext | mandatory | string | error description |
| result | mandatory |  | API result |

result

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| LastDate | mandatory | datetime | last request datetime |
| betId | mandatory | long | bet no |
| account | mandatory | varchar(32) | member’s account |
| betAmount | mandatory | decimal | betAmount |
| betOdd | mandatory | decimal | odds |
| win | mandatory | decimal | win / lose |
| comm | mandatory | decimal | commission |
| betType | mandatory | varchar(2) | refer to 7.1.6 sports bet type |
| oddStyle | mandatory | varchar(1) | E: Europe  H: HongKong  M: Malysia  I: Indunisia |
| hdp | mandatory | decimal | handicap |
| betPos | mandatory | varchar(32) | Bet Position (Home/Away;Over/Under;Odd/Even...) |
| live | mandatory | Boolean | is live bet |
| betDate | mandatory | datetime |  |
| status | mandatory | varchar(32) | bet status  rejected  running  settled |
| result | mandatory | varchar(2) | Win  Lose  Win/2  Lose/2  Draw |
| reportDate | mandatory | datetime |  |
| sportName | mandatory | varchar(32) | sport type name |
| leagueId | mandatory | int | league |
| homeId | mandatory | int | team |
| awayId | mandatory | int | team |
| betScore | mandatory | varchar(32) | match score when bet |
| matchDate | mandatory | datetime | match start time |
| betIp | mandatory | varchar(32) |  |
| matchId | mandatory | int | match id |
| updateTime | mandatory | datetime | bet update time |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion="1.0"encoding="utf-8"?>  <response>  <errcode>0</errcode>  <errtext> </errtext>  <result>  <lastdate>2013-08-27 10:05:35</lastdate>  <betlist>  <bet>  <BetID>4775749</BetID>  <Account>zzzaaaaaatj</Account>  <BetAmount>110.0000</BetAmount>  <BetOdd>73.8153</BetOdd>  <Win>0.0000</Win>  <Comm>0.0000</Comm>  <BetType>Mix Parlay</BetType>  <OddStyle>E</OddStyle>  <Hdp>0.0000</Hdp>  <BetPos>4X11</BetPos>  <Live>0</Live>  <BetDate>2013-08-27 10:05:35</BetDate>  <Status>Running</Status>  <Result></Result>  <ReportDate>2013-08-27 00:00:00</ReportDate>  <SportName>Soccer</SportName>  <LeagueID>0</LeagueID>  <HomeID>0</HomeID>  <AwayID>0</AwayID>  <BetScore>0-0</BetScore>  <MatchDate></MatchDate>  <BetIP>123.145.173.181</BetIP>  <MatchID>0</MatchID>  <UpdateTime>2013-08-27 23:05:03</UpdateTime>  </bet>  </betlist>  </result>  </response> |

* + 1. GetBetSheetByReport

**Mount Point** ~/GetBetSheetByReport?APIPassword={**apiPassword**}&MemberAccount={**memberAccount}&**BetId={**betId**}&Status={**status**}&ReportDate={**reportDate**}

The API is used to get gspMember’s bet limit information.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | varchar(32) | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| memberAccount | mandatory | string | gspMember id |
| betId | mandatory | long | bet no |
| status | mandatory | int | 0:ALL  1:Running  2:Settled  3:Rejected |
| reportDate | mandatory | datetime | yyyy-MM-dd HH:mm:ss |

Example

|  |
| --- |
| http://domain/sportapi.asmx/GetBetSheetByReport?APIPassword=f0a3ae129d0685a3073c7c6f5bcc6107&MemberAccount=&BetID=0&Status=-1&ReportDate=2013-08-15 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code |
| errtext | mandatory | string | error description |
| result | mandatory |  | API result |

result

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| LastDate | mandatory | datetime | last request datetime |
| betId | mandatory | long | bet no |
| account | mandatory | varchar(32) | member’s account |
| betAmount | mandatory | decimal | betAmount |
| betOdd | mandatory | decimal | odds |
| win | mandatory | decimal | win / lose |
| comm | mandatory | decimal | commission |
| betType | mandatory | varchar(2) | refer to 7.1.6 sports bet type |
| oddStyle | mandatory | varchar(1) | E: Europe  H: HongKong  M: Malysia  I: Indunisia |
| hdp | mandatory | decimal | handicap |
| betPos | mandatory | varchar(32) | Bet Position (Home/Away;Over/Under;Odd/Even...) |
| live | mandatory | Boolean | is live bet |
| betDate | mandatory | datetime |  |
| status | mandatory | varchar(32) | bet status  rejected  running  settled |
| result | mandatory | varchar(2) | Win  Lose  Win/2  Lose/2  Draw |
| reportDate | mandatory | datetime |  |
| sportName | mandatory | varchar(32) | sport type name |
| leagueId | mandatory | int | league |
| homeId | mandatory | int | team |
| awayId | mandatory | int | team |
| betScore | mandatory | varchar(32) | match score when bet |
| matchDate | mandatory | datetime | match start time |
| betIp | mandatory | varchar(32) |  |
| matchId | mandatory | int | match id |
| updateTime | mandatory | datetime | bet update time |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion="1.0"encoding="utf-8"?>  <response>  <errcode>0</errcode>  <errtext> </errtext>  <result>  <lastdate>2013-08-27 10:05:35</lastdate>  <betlist>  <bet>  <BetID>4775749</BetID>  <Account>zzzaaaaaatj</Account>  <BetAmount>110.0000</BetAmount>  <BetOdd>73.8153</BetOdd>  <Win>0.0000</Win>  <Comm>0.0000</Comm>  <BetType>Mix Parlay</BetType>  <OddStyle>E</OddStyle>  <Hdp>0.0000</Hdp>  <BetPos>4X11</BetPos>  <Live>0</Live>  <BetDate>2013-08-27 10:05:35</BetDate>  <Status>Running</Status>  <Result></Result>  <ReportDate>2013-08-27 00:00:00</ReportDate>  <SportName>Soccer</SportName>  <LeagueID>0</LeagueID>  <HomeID>0</HomeID>  <AwayID>0</AwayID>  <BetScore>0-0</BetScore>  <MatchDate></MatchDate>  <BetIP>123.145.173.181</BetIP>  <MatchID>0</MatchID>  <UpdateTime>2013-08-27 23:05:03</UpdateTime>  </bet>  </betlist>  </result>  </response> |

* + 1. GetParlayBetSheet

**Mount Point** ~/ GetParlayBetSheet?APIPassword={**apiPassword**}&MemberAccount={**memberAccount}&**BetId={**betId**}&Status={**status**}&FromDate={**fromDate}&ToDate={toDate}**

The API is used to get gspMember’s bet limit information.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | varchar(32) | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| memberAccount | mandatory | string | gspMember id |
| betId | mandatory | long | bet no |
| fromDate | mandatory | datetime | yyyy-MM-dd HH:mm:ss |
| toDate | mandatory | datetime | yyyy-MM-dd HH:mm:ss |

Example

|  |
| --- |
| http://domain/sportapi.asmx/GetBetSheetByReport?APIPassword=f0a3ae129d0685a3073c7c6f5bcc6107&MemberAccount=&BetID=0&Status=-1&ReportDate=2013-08-15 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code |
| errtext | mandatory | string | error description |
| result | mandatory |  | API result |

result

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| LastDate | mandatory | datetime | last request datetime |
| betId | mandatory | long | bet no |
| no | mandatory | long | mix parlay no |
| betOdd | mandatory | decimal | odds |
| betType | mandatory | varchar(2) | refer to 7.1.6 sports bet type |
| hdp | mandatory | decimal | handicap |
| betPos | mandatory | varchar(32) | Bet Position (Home/Away;Over/Under;Odd/Even...) |
| status | mandatory | varchar(32) | bet status  rejected  running  settled |
| result | mandatory | varchar(2) | Win  Lose  Win/2  Lose/2  Draw |
| sportName | mandatory | varchar(32) | sport type name |
| leagueId | mandatory | int | league |
| homeId | mandatory | int | team |
| awayId | mandatory | int | team |
| matchDate | mandatory | datetime | match start time |
| matchId | mandatory | int | match id |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion="1.0"encoding="utf-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result>  <lastdate>2013-08-27 10:05:35</lastdate>  <betlist>  <bet>  <BetID>4775749</BetID>  <No>390987</No>  <BetOdd>2.0300</BetOdd>  <BetType>First 1X2</BetType>  <Hdp>0.0000</Hdp>  <BetPos>Away</BetPos>  <Status>Running</Status>  <Result></Result>  <SportName>Soccer</SportName>  <LeagueID>14367</LeagueID>  <HomeID>32237</HomeID>  <AwayID>32659</AwayID>  <MatchDate>2013-08-28 02:45:00</MatchDate>  <MatchID>579834</MatchID>  </bet>  </betlist>  </result>  </response> |

* + 1. GetLeagueName

**Mount Point** **~/GetLeagueName?APIPassword={apiPassword}&LeagueID={leagueId }&MinLeagueID={minLeagueId}**

The API is used to get name of league(?)

Notice:Return 50 records every requst;

Please back last LeagueID and MinLeagueID=true

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | string | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| leagueID | mandatory | string | league id |
| minLeagueId | mandatory | int | minimum league id  If you want to get one league's name  then set LeagueID>0 and set MinLeagueID=true;  If you want to get all team's name then  set LeagueID=0 and set MinLeagueID=false; |

Example

|  |
| --- |
| http://domain/sportapi.asmx/GetLeagueName?APIPassword=f0a3ae129d0685a3073c7c6f5bcc6107&LeagueID=0&MinLeagueID=false |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code |
| errtext | mandatory | string | error description |
| result | mandatory | varchar(32) | API result |

result

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| LeagueId | mandatory | long | league id |
| EN | mandatory | varchar(255) | English |
| CH | mandatory | varchar(255) | Chinese |
| TW | mandatory | varchar(255) | Traditional Chinese |
| VN | mandatory | varchar(255) | Vietnam |
| TH | mandatory | varchar(255) | Thailand |
| KH | mandatory | varchar(255) | Cambodia |
| KO | mandatory | varchar(255) | Korea |
| JP | mandatory | varchar(255) | Japanese |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion="1.0"encoding="utf-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result>  <league>  <LeagueID>5449</LeagueID>  <EN>Portugal SuperLiga</EN>  <CH>葡萄牙超级联赛</CH>  <TW>葡萄牙超级联赛</TW>  <VN>Portugal SuperLiga</VN>  <TH>Portugal SuperLiga</TH>  <KH>Portugal SuperLiga</KH>  <KO>포르투갈슈퍼리가</KO>  <JP>ポルトガルスーパーリーグ</JP>  </league>  <league>  <LeagueID>5450</LeagueID>  <EN>Italy Serie B</EN>  <CH>意大利乙组联赛</CH>  <TW>意大利乙组联赛</TW>  <VN>Italy Serie B</VN>  <TH>อิตาลีซีรีย์บี</TH>  <KH>Italy Serie B</KH>  <KO>이탈리아세리에 B</KO>  <JP>イタリアセリエB</JP>  </league>  </result>  </response> |

* + 1. GetTeamName

**Mount Point** ~/GetTeamName?APIPassword={**apiPassword**}&TeamID={**teamId**}&MinTeamId={**minTeamId**}

The API is used to get a team name.

Notice:Return 50 records every requst;Please back last LeagueID and MinLeagueID=true

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | string | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| teamId | mandatory | string | team id |
| minTeamId | mandatory | boolean | minimum team id  If you want to get one team's name then  set TeamID>0 and set MinTeamID=true;  If you want to get all team'name then  set TeamID=0 and set MinTeamID=false; |

Example

|  |
| --- |
| http://domain/sportapi.asmx/GetTeamName?APIPassword=f0a3ae129d0685a3073c7c6f5bcc6107&TeamID=0&MinTeamID=false |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code |
| errtext | mandatory | string | error description |
| result | mandatory | varchar(32) | API result |

result

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| TeamId | mandatory | long | league id |
| EN | mandatory | varchar(255) | English |
| CH | mandatory | varchar(255) | Chinese |
| TW | mandatory | varchar(255) | Traditional Chinese |
| VN | mandatory | varchar(255) | Vietnam |
| TH | mandatory | varchar(255) | Thailand |
| KH | mandatory | varchar(255) | Cambodia |
| KO | mandatory | varchar(255) | Korea |
| JP | mandatory | varchar(255) | Japanese |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion="1.0"encoding="utf-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result>  <team>  <TeamID>31894</TeamID>  <EN>TSV 1860 Munchen Am</EN>  <CH>1860慕尼黑Am</CH>  <TW>1860慕尼黑Am</TW>  <VN>TSV 1860 Munchen Am</VN>  <TH>ทีเอสวี 1860 มูเชนแอม</TH>  <KH>TSV 1860 Munchen Am</KH>  <KO>TSV 1860 Munchen Am</KO>  <JP>TSV1860 ミュンヘンAm</JP>  </team>  <team>  <TeamID>31895</TeamID>  <EN>TSV 1860 Munchen</EN>  <CH>慕尼黑1860</CH>  <TW>慕尼黑1860</TW>  <VN>TSV 1860 Munchen</VN>  <TH>ทีเอสวี 1860 มูเชิน</TH>  <KH>TSV 1860 Munchen</KH>  <KO>TSV 1860 뮌헨</KO>  <JP>TSV 1860ミュンヘン</JP>  </team>  </result>  </response> |

* + 1. GetMatchResult

**Mount Point** ~/GetMatchResult?APIPassword={**apiPassword**}&LeagueID={**leagueId**}&MinLeagueID={**minLeagueId**}

The API is used to get a result of a match.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | string | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| matchID | mandatory | string | match id: specific sports game id |

Example

|  |
| --- |
| http://domain/sportapi.asmx/GetMatchResult?APIPassword=f0a3ae129d0685a3073c7c6f5bcc6107&MatchID=574252 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code |
| errtext | mandatory | string | error description |
| result | mandatory | varchar(32) | API result |

result

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| HalfScore | mandatory | varchar(32) | Home score : Away Score at HalfTIme |
| FullScore | mandatory | varchar(32) | Home Score : Away Score at FullTime |
| HomeCard | mandatory | varchar(32) | Red card of home |
| AwayCard | mandatory | varchar(32) | Red card of away |
| FirstGoal | mandatory | varchar(32) | Who( home / away ) first goal |
| LastGoal | mandatory | varchar(32) | Who( home / away ) last goal |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion="1.0"encoding="utf-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result>  <HalfScore>1-2</HalfScore>  <FullScore>3-3</FullScore>  <HomeCard>0</HomeCard>  <AwayCard>0</AwayCard>  <FirstGoal>Home</FirstGoal>  <LastGoal>Away</LastGoal>  </result>  </response> |

* + 1. UpdateGroupComm

**Mount Point** ~/GetMatchResult?APIPassword={**apiPassword**}&MemberAccount={**memberAccount** }&GroupComm ={**groupComm**}

The API is used to update group commission.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | varchar(32) | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| memberAccount | mandatory | varchar(32) | gspMember id |
| groupComm | mandatory | varchar(32) | commission group |

Example

|  |
| --- |
| http://domain/sportapi.asmx/UpdateGroupComm?APIPassword=f0a3ae129d0685a3073c7c6f5bcc6107&MemberAccount=test&GroupComm=A |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code |
| errtext | mandatory | string | error description |
| result | mandatory | varchar(32) | API result |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion="1.0"encoding="utf-8"?>  <response>  <errcode>000000</errcode>  <errtext></errtext>  <result></result>  </response> |

* + 1. UpdateStatus

**Mount Point** ~/UpdateStatus?APIPassword={**apiPassword**}&MemberAccount={**memberAccount** }&GroupComm ={**groupComm**}

The API is used to update a member’s status.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | varchar(32) | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| memberAccount | mandatory | varchar(32) | gspMember id |
| Closed | mandatory | boolean | Can’t login  true:close  false:open |
| Paused | mandatory | Boolean | Can Login, Can’t Bet  true:pause  false:open |

Example

|  |
| --- |
| http://domain/sportapi.asmx/UpdateStatus?APIPassword=f0a3ae129d0685a3073c7c6f5bcc6107&MemberAccount=test&Cloesd=true&Paused=true |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code |
| errtext | mandatory | string | error description |
| result | mandatory | varchar(32) | API result |

Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion="1.0"encoding="utf-8"?>  <response>  <errcode>000000</errcode>  <errtext></errtext>  <result></result>  </response> |

* + 1. GetStatus

**Mount Point** ~/GetStatus?APIPassword={**apiPassword**}&MemberAccount={**memberAccount** }

The API is used to get a member’s status.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| apiPassword | mandatory | varchar(32) | api password  dad756r5a8511hb72c6dgf0f4f193cs5 |
| memberAccount | mandatory | varchar(32) | gspMember id |

Example

|  |
| --- |
| http://domain/sportapi.asmx/GetStatus?APIPassword=f0a3ae129d0685a3073c7c6f5bcc6107&MemberAccount=test |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code |
| errtext | mandatory | string | error description |
| result | mandatory | varchar(32) | API result |

result

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Closed | mandatory | Boolean | Can’t Login  true:close  false:open |
| Paused | mandatory | Boolean | Can Login, Can’t Bet  true:pause  false:open |

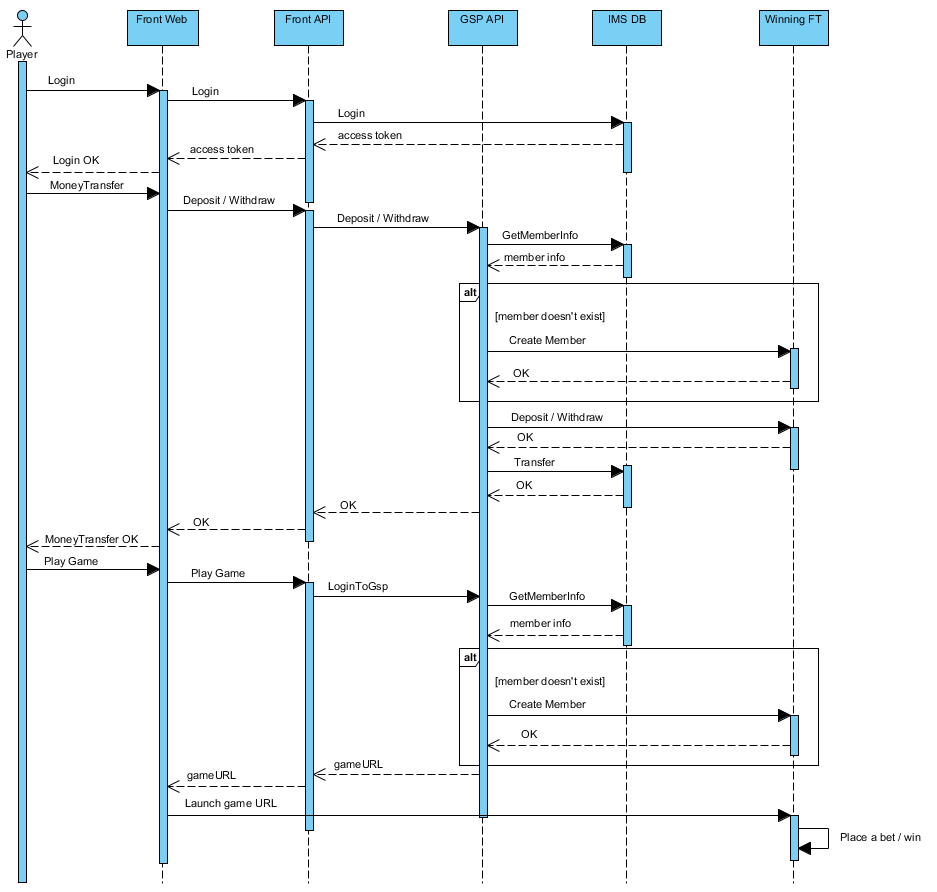
Response Codes

|  |  |  |
| --- | --- | --- |
| **HTTP RESPONSE** | **STATUS** | **DESCRIPTION** |
| 200 | OK |  |
| 400 | BAD REQUEST |  |

Example

|  |
| --- |
| <?xmlversion="1.0"encoding="utf-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result>  <Closed>true</Closed>  <Paused>true</Paused>  </result>  </response> |

* 1. M8 (M8)



|  |  |  |  |
| --- | --- | --- | --- |
| **Reference Document** | **Version** | **File Name** | **API** |
| Sportsbook API |  | Sportbooks API M8 - 2014-11-18 (1).docx |  |

General Error Codes

|  |  |
| --- | --- |
| **Error ID** | **Error Description** |
| -1 | System maintenance |
| -2 | Invalid secret |
| -3 | Ip address not allow |
| -4 | Invalid username (length 1-14, format A-Za-z0-9)"; |
| -5 | Invalid action |
| -7 | Invalid agent |
| -8 | Invalid username"; |
| -100 | Invalid Intput |
| -101 | Exception |

**connectionURL: merchPassword2 of tb\_game\_list**

* + 1. CreateMember (GET)

**Mount Point:** {**connectionURL**}?agent={agent}&secret={secret}&action=create&username={username}&currency={currency}

The API is used to create new member account on Winning Sports Game Service Provider.

***Request***

Request Parameters

**connectionURL: merchPassword2 of tb\_game\_list**

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | 1jm1 |
| secret | mandatory | string | m75139b |
| action | mandatory | string | create |
| username | mandatory | string | gspMember id |
| currency | mandatory | string | currency ISO code |

Example

|  |
| --- |
| http://apiweb.mywinday.com/api.aspx?agent=1jm1&secret=m75139b&action=create&username=test100&currency=USD |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  1: Username exist  2: Invalid currency  + general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | gspMember id |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result>test100</result>  </response> |

* + 1. Update (GET)

**Mount Point**:

{**connectionURL**}?agent={agent}&secret={secret}&action=**update**&username={username}&max1={max1}& max2={max2}&max3={max3}&max4={max4}&lim1={lim1}&lim2={lim2}&lim3={lim3}&comtype ={comtype}&com1={com1}&com2={com2}&com3={com3}&suspend={suspend}

The API is used to update a member’s max bets, limits and commission settings.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | 1jm1 |
| secret | mandatory | string | m75139b |
| action | mandatory | string | update |
| username | mandatory | string | gspMember id |
| max1 | mandatory | string | Max bet for HDP/OU/OE |
| max2 | mandatory | string | Max bet for 1X2 |
| max3 | mandatory | string | Max bet for PAR |
| max4 | mandatory | string | Max bet for ORT/CS/TG/HFT/FLG |
| lim1 | mandatory | string | Per match limit for all except PAR/ORT |
| lim2 | mandatory | string | Per match payout limit for PAR (payout = bet \* odds) |
| lim3 | mandatory | string | Per match limit for ORT |
| lim4 | mandatory | string |  |
| comtype | mandatory | string | Choice of A,B,C,D,E,F,4 for HDP/OU/OE |
| com1 | mandatory | string | Commission for HDP/OU/OE |
| com2 | mandatory | string | Commisson for 1X2/ORT |
| com3 | mandatory | string | Commission for PAR/CS/TG/HFT/FLG |
| suspend | mandatory | string | 0: no suspend, 1:suspend |

Example

|  |
| --- |
| http://apiweb.mywinday.com/api.aspx?agent=1jm1&secret=m75139b&action=update&username=test100&&max1=1100& max2=1200&max3=1300&max4=1400&lim1=1000&lim2=1100&lim3=1200&comtype=A&com1=0.1&com2=0.2&com3=0.3&suspend=0 |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error |
| errtext | mandatory | string | error message |
| result | mandatory | string | gspMember id |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>1</errcode>  <errtext>EXCEED</errtext>  <result></result>  </response> |

* + 1. CheckBalance (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=balance&username={username}

The API is used to look up current balance of a member.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | 1jm1 |
| secret | mandatory | string | m75139b |
| action | mandatory | string | balance |
| username | mandatory | string | gspMember id |

Example

|  |
| --- |
| http://apiweb.mywinday.com/api.aspx?agent=1jm1&secret=m75139b&action=balance&username=test100 |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  1: User not exist  2: Account locked  3: System maintainenace  4: Egame online  + general error |
| errtext | mandatory | string | error message |
| result | mandatory | string | balance |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result>0</result>  </response> |

* + 1. Deposit

**Mount Point:**

{**connectionURL**}?agent={agent}&secret={secret}&action=deposit&username={username}&amount={amount}&serial={ serial}

The API is used to perform fund in.

You must call "Check Balance" with no error before deposit.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | 1jm1 |
| secret | mandatory | string | m75139b |
| action | mandatory | string | deposit |
| username | mandatory | string | gspMember id |
| amount | mandatory | string | transaction amount |
| serial | mandatory | string | transaction id |

Example

|  |
| --- |
| http://apiweb.mywinday.com/api.aspx?agent=1jm1&secret=m75139b&&action=deposit&username=test100&amount=100&serial=100 |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code |
| errtext | mandatory | string | error message |
| result | mandatory | string | result of API |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result>100</result>  </response> |

* + 1. Withdraw (GET)

**Mount Point:** {**connectionURL**}?agent={agent}&secret={secret}&action=withdraw&username={username}&amount={amount}&serial={serial}

The API is used to perform fund out.

You must call "Check Balance" with no error before deposit.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | 1jm1 |
| secret | mandatory | string | m75139b |
| action | mandatory | string | withdraw |
| username | mandatory | string | gspMember id |
| amount | mandatory | string | transaction amount |
| serial | mandatory | string | transaction id |

Example

|  |
| --- |
| http://apiweb.mywinday.com/api.aspx?agent=1jm1&secret=m75139b&&action=withdraw&username=test100&amount=100&serial=101 |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code |
| errtext | mandatory | string | error message |
| result | mandatory | string | result of API |

Example

|  |  |
| --- | --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result>0</result>  </response> |  |

* + 1. Login (GET)

**Mount Point**:

{**connectionURL**}?agent={agent}&secret={secret}&action=login&username={**username**}&lang=EN-US&host={**host**}

The API is used to login the game

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | 1jm1 |
| secret | mandatory | string | m75139b |
| action | mandatory | string | login |
| username | mandatory | string | gspMember id |
| lang | mandatory | string | Web Browser Language Identification Codes  EN-US(hard-code) |
| host | mandatory | string | game url |

Example

|  |
| --- |
| http://apiweb.mywinday.com/api.aspx?agent=1jm1&secret=m75139b&action=login&username=test100&lang=EN-US&host=sport.mywinday.com |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  1: Internal error: invalid LoginPrivateKey |
| errtext | mandatory | string | error message |
| result | mandatory | string | User login url |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>-10</errcode>  <errtext>Invalid host (sp4x4avw)</errtext>  <result></result>  </response> |

* + 1. Logout (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=logout&username={**username**}

The API is used to logout the game.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | 1jm1 |
| secret | mandatory | string | m75139b |
| action | mandatory | string | logout |
| username | mandatory | string | gspMember id |

Example

|  |
| --- |
| http://apiweb.mywinday.com/api.aspx?agent=1jm1&secret=m75139b&action=logout&username=test100 |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : no error |
| errtext | mandatory | string | error message |
| result | mandatory | string |  |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>0</errcode>  <errtext> </errtext>  <result></result>  </response> |

* + 1. CheckPayment (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=check\_payment&username={**username**}&serial ={**serial**}

The API is used to check gspMember’s payment.

If username is empty, check all users.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | 1jm1 |
| secret | mandatory | string | m75139b |
| action | mandatory | string | check\_payment |
| username | mandatory | string | gspMember id |
| serial | optional | string | serial |

Example

|  |
| --- |
| http://apiweb.mywinday.com/api.aspx?agent=1jm1&secret=m75139b&action=check\_payment&username=test100&serial=100 |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | List<Payment> |

Payment

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Username | mandatory | string | gspMember id |
| Amount | mandatory | string | ?? |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result>  <payment>  <username>1jm1test100</username>  <amount>100</amount>  </payment>  </result>  </response> |

* + 1. CheckOnlineUsers (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=check\_online&username={**username**}

The API is used to check online members.

If username is empty, check all users.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | 1jm1 |
| secret | mandatory | string | m75139b |
| action | mandatory | string | check\_online |
| username | mandatory | string | gspMember id |

Example

|  |
| --- |
| http://apiweb.mywinday.com/api.aspx?agent=1jm1&secret=m75139b&action=check\_online&username=test100 |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | List<string>: list of username (gspMember id) |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result></result>  </response> |

* + 1. Ticket (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=ticket&username={**username**}&start ={**start**}& duration={**duration**}&match\_over={**match\_over**}

The API is used to ???

Date and time range is based ticket transaction date.

Maximum 100 tickets can be returned.

If username is empty, check all users.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | 1jm1 |
| secret | mandatory | string | m75139b |
| action | mandatory | string | ticket |
| username | mandatory | string | gspMember id |
| start | mandatory | string | start date time  datetime format?? |
| duration | mandatory | string | number of seconds |
| match\_over | mandatory | string | 0 or 1 |

Example

|  |
| --- |
| http://apiweb.mywinday.com/api.aspx?agent=1jm1&secret=m75139b&action=ticket&username=test100&start=2015-08-18+09%3A10%3A00&duration=600&match\_over=0 |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | List<Ticket> |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result></result>  </response> |

* + 1. Parlay (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=parlay&username={**username**}&ticket\_id={**ticket\_id**}

The API is used to ???

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | 1jm1 |
| secret | mandatory | string | m75139b |
| action | mandatory | string | parlay |
| username | mandatory | string | gspMember id |
| ticket\_id | mandatory | string | ticket\_id |

Example

|  |
| --- |
| http://apiweb.mywinday.com/api.aspx?agent=1jm1&secret=m75139b&action=parlay&username=test100&ticket\_id=321232 |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | List<Ticket> |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result></result>  </response> |

* + 1. Team (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=team&username={**username**}&team\_id={**team\_id**}

The API is used to get an information of searching team.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | 1jm1 |
| secret | mandatory | string | m75139b |
| action | mandatory | string | team |
| username | mandatory | string | gspMember id |
| team\_id | mandatory | string | team\_id |

Example

|  |
| --- |
| http://apiweb.mywinday.com/api.aspx?agent=1jm1&secret=m75139b&action=team&username=test100&team\_id=manu |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | <name><lang>en-US</lang><txt>Arsenal</txt></name>... |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>-100</errcode>  <errtext>Invalid team\_id: manu</errtext>  <result></result>  </response> |

* + 1. League (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=league&username={**username**}&league\_id ={**league\_id**}

The API is used to ???

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | 1jm1 |
| secret | mandatory | string | m75139b |
| action | mandatory | string | league |
| username | mandatory | string | gspMember id |
| league\_id | mandatory | string | league\_id |

Example

|  |
| --- |
| http://apiweb.mywinday.com/api.aspx?agent=1jm1&secret=m75139b&action=league&username=test100&league\_id=uefa |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | <name><lang>en-US</lang><txt>UEFA</txt></name>... |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>-100</errcode>  <errtext>Invalid league\_id: uefa</errtext>  <result></result>  </response> |

* + 1. Fetch (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=fetch&username={**username**}

The API is used to ???

Maximum 100 tickets can be returned.

Tickets will be available for fetch when ticket is :

(a) Accepted (b) Rejected (c) Cancelled (d) Match over.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | 1jm1 |
| secret | mandatory | string | m75139b |
| action | mandatory | string | fetch |
| username | mandatory | string | gspMember id |

Example

|  |
| --- |
| http://apiweb.mywinday.com/api.aspx?agent=1jm1&secret=m75139b&action=fetch&username=test100 |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | List<Ticket> |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result></result>  </response> |

* + 1. Mark Fetched (GET)

**Mount Point**: {**connectionURL**}?agent={agent}&secret={secret}&action=mark\_fetched&username={**username**}& fetch\_ids={**fetch\_ids**}

The API is used to ???

Mark tickets as fetched so that it will not be fetched again.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| agent | mandatory | string | 1jm1 |
| secret | mandatory | string | m75139b |
| action | mandatory | string | mark\_fetched |
| username | mandatory | string | gspMember id |
| fetch\_ids | mandatory | string | fetch\_ids |

Example

|  |
| --- |
| http://apiweb.mywinday.com/api.aspx?agent=1jm1&secret=m75139b&action=mark\_fetched&username=test100&fetch\_ids=12321,2412 |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| errcode | mandatory | string | error code  : general error codes |
| errtext | mandatory | string | error message |
| result | mandatory | string | none |

Example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <response>  <errcode>0</errcode>  <errtext></errtext>  <result></result>  </response> |

* 1. XTD (XTD) – moved to SingleWallet API

Document Version: LiveGame API User Manual v1.1.031

( File name: API USER MANUAL v1.1.031 EN.pdf)

* + 1. CreateMember

Mount Point: ~/register

The API is used to create new member account on XTD Game Service Provider.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| username | mandatory | varchar(50) | gspMember id |
| password | mandatory | varchar(50) | should be encrypted by MD5.  but don’t in source code.  hard-code: 12345 |
| agentCode | mandatory | string | gspGameCurrencyId |

The form of the input data: username|12345|agentCode;

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| ReturnCode | mandatory | int | Return Code |

Example

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **RETURN CODE** | **DESCRIPTION** |
| 0 | the register is successful |
| 1 | the register is failed |
| 2 | length of username and password exceed 50 characters |
| 3 | other errros |

* + 1. GetMemberBalance

Mount Point: ~/checkBalance

The API is used to get a member’s balance on XTD Game Service Provider.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| username | mandatory | varchar(50) | gspMember id |
| agentCode | mandatory | string | gspGameCurrencyId |

The form of the input data: username + “|” + agentCode;

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| ReturnCode | mandatory | string | Return Code  -1: the operation is failed  other numbers: balance |

Example

|  |
| --- |
|  |

* + 1. Credit

Mount Point: ~/transfer

The API is used to transfer credit from operator account to game service provider account

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| username | mandatory | varchar(50) | gspMember id |
| agentCode | mandatory | string | gspGameCurrencyId |
| amount | mandatory | decimal | transaction amount  a positive number: credit  a negative number: debit |
| tranRefID | optional | string | Only at the times when the database has no record same as the tranRefID, the transfer can be successful. |

The form of the input data: username|agentCode|amount

The form of the input data: username|agentCode|amount|tranRefID

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| ReturnCode | mandatory | int | Return Code |

Example

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **RETURN CODE** | **DESCRIPTION** |
| 0 | the transfer is successful |
| 2 | Not enough amount |
| 3 | other errros |
| 4 | User Not found |
| 5 | Transaction already exist |
| 6 | Invalid amount |
| 7 | Invalid prefer code |
| 11 | the amount is decimal (only integer is allowed) |

* + 1. Debit

Mount Point: ~/transfer

The API is used to transfer credit from game service provider account to operator account.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| username | mandatory | string | gspMember id |
| agentCode | mandatory | string | gspGameCurrencyId |
| amount | mandatory | decimal | transaction amount  a positive number: credit  a negative number: debit |
| tranRefID | optional | string | Only at the times when the database has no record same as the tranRefID, the transfer can be successful. |

The form of the input data: username|agentCode|amount

The form of the input data: username|agentCode|amount|tranRefID

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| ReturnCode | mandatory | int | Return Code |

Example

|  |
| --- |
|  |

* + 1. PlayGame

Mount Point: ~/login

The API is used to play a XTD game.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| username | mandatory | string | gspMember id |
| userpassword | mandatory | string | should be encrypted by MD5.  it is hard-coded in source code.  default: 12345 |
| agentCode | mandatory | string | opToGspCurrencyId |
| language | mandatory | int | 1: Chinese  2: English  3: Korean |
| autoCreate | optional | int | 1: auto register function can be used  0 or other number: cannot be used |

The form of the input data: username|userpassword|agentCode|language

The form of the input data: username|userpassword|agentCode|language|autoCreate

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| ReturnCode | mandatory | int | return code |
| ReturnMessage | mandatory | string | return message |

Example

|  |
| --- |
|  |

* + 1. GetJackpotPool (Unused)

check source.

method header: cGetJackpotPool , same as GD?

* 1. SBO (SBO)
     1. RegisterPlayer (GET)

**Mount Point:** http://103.9.150.43/sportsfundservice/registerplayer/{**opcode**}/player/{**username**}/agent/{**agent**}/lang/{**language**}/{\***country**}?access={**access**}

The API is used to create new member account on SBO Game Service Provider.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| opcode | mandatory | string(10) | The operator code provided by SBO. |
| username | mandatory | string(20) | opcode + player Id |
| agent | mandatory | string(20) | The username of agent which this player account is under. |
| language | mandatory | string | Browser Language code  en (default)  zh-tw  zh-cn  th-th  Id-id |
| country | optional | string(2) | country ISO 3166 alpha-2 |
| access | mandatory | string(32) | MD5 Hashing (take the path from after the ‘sportsfundservice/’ + ‘/’ + Private Key |

Example

|  |
| --- |
| http://103.9.150.43/sportsfundservice/registerplayer/M777/player/aaaa32/agent/aaa18/lang/en?access=5848a68d6e22308b60ebfdb6c51de4de |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error | mandatory | Error | error |
| serverId | mandatory | string | server id |

Error

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| id | mandatory | string | error id |
| msg | mandatory | string | error message |

Example

|  |
| --- |
| {  "error": {  "id": 4102,  "msg": "CreatePlayerFail"  },  "serverId": "A01"  } |

* + 1. Login(GET)

**Mount Point:** http://103.9.150.43/sportsfundservice/login/{**opcode**}/player/{**username**}?access={**access**}

The API is used to login on the SBO system.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| opcode | mandatory | string(10) | The operator code provided by SBO. |
| username | mandatory | string(20) | opcode + player Id |
| access | mandatory | string(32) | MD5 Hashing (take the path from after the ‘sportsfundservice/’ + ‘/’ + Private Key |

Example

|  |
| --- |
| http://103.9.150.43/sportsfundservice/login/M777/player/aaaa32?access=4b4cfe29966f26e8af33e817f0ecec281 |

***Response***

**If No error, API return the login url with token.**

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error | mandatory | Error | error |
| serverId | mandatory | string | server id |
| username | mandatory | string | request username |
| token | mandatory | string | sbo token |

Error

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| id | mandatory | string | error id |
| msg | mandatory | string | error message |

Example

|  |
| --- |
| {  “username” : “aaa01”,  “token” : “21321.JKDFAxk217daJLKdfa”  “error” : { “id” : 0, “msg” : “no error”},  “serverId” : ‘RSP-07’  } |

* + 1. UpdatePlayerStatus(GET)

**Mount Point:** http://103.9.150.43/sportsfundservice/updateplayerstatus/{**opcode**}/player/{**username**}/status/{**status**}?access={**access**}

The API is used to login on the SBO system.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| opcode | mandatory | string(10) | The operator code provided by SBO.  GBT88 |
| username | mandatory | string(20) | opcode + player Id |
| status | mandatory | string | active:The account is active means he could perform all actions normally  suspend: A suspend account is not able to place order in system. All other actions are able to preform normally  closed: A closed account is not able to do any action in system including login. If the account is logged in, he will be kick out. |
| access | mandatory | string(32) | MD5 Hashing (take the path from after the ‘sportsfundservice/’ + ‘/’ + Private Key |

Example

|  |
| --- |
| http://103.9.150.43/sportsfundservice/updateplayerstatus/M777/player/aaaa/status/active?access=c18f153141eceea989398c473df2d11fd |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error | mandatory | Error | error |
| serverId | mandatory | string | server id |

Error

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| id | mandatory | string | error id |
| msg | mandatory | string | error message |

Example

|  |
| --- |
| {  “error” : { “id” : 0, “msg” : “no error”},  “serverId” : ‘RSP-07’  } |

* + 1. Deposit(GET)

**Mount Point:** http://103.9.150.43/sportsfundservice/deposit/{**opcode**}/player/{**username**}/amount/{**amount**}/txnid/{**txnId**}?access={**access**}

This API is for player to transfer money from IMS main wallet into SBO system. SBO system only allow player deposit on his own currency defined when registration.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| opcode | mandatory | string(10) | The operator code provided by SBO.  GBT88 |
| username | mandatory | string(20) | opcode + player Id |
| amount | mandatory | decimal | decimal value: up to 2 decimal points.  Eg. 12.21, 9 |
| txnId | mandatory | string(19) | It must be unique among all transactions (deposit/withdraw) in the IMS.  All txnId is prepend with opcode. |
| access | mandatory | string(32) | MD5 Hashing (take the path from after the ‘sportsfundservice/’ + ‘/’ + Private Key |

Example

|  |
| --- |
| http://103.9.150.43/sportsfundservice/deposit/M777/player/aaaa32/amount/10000.00/txnid/aaaa32-15?access=aa3a4f52113c727170178b7a2aa73b74 |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error | mandatory | Error | error |
| serverId | mandatory | string | server id |
| txnId | mandatory | string(30) | prepend with opcode and “-“ |
| refno | mandatory | string(64) | generated by SBO system and unique among all transactions. |
| balance | mandatory | decimal | player Gsp Balance |

Error

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| id | mandatory | string | error id |
| msg | mandatory | string | error message |

Example

|  |
| --- |
| {  “error” : { “id” : 0, “msg” : “no error”},  “serverId” : ‘RSP-07’  } |

* + 1. Withdraw (GET)

**Mount Point:** http://103.9.150.43/sportsfundservice/withdrawal/{**opcode**}/player/{**username**}/amount/{**amount**}/txnid/{**txnId**}/{\***fullamount**}?access={**access**}

This API is for player to transfer money from SBO System into IMS main wallet.

SBO system only allow player withdraw on his own currency defined when registration.

Transaction Id and reference no is one to one mapping. If two request with same transaction id, the 2nd request will return error.

Player is only allowed to withdraw available balance.

If there are pending orders, player is not able to withdraw the wager until it is settled.

Player is only allowed to withdraw up to cents level. If player’s balance is 19.4297 in white label system, his withdrawal max is capped at 19.42.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| opcode | mandatory | string(10) | The operator code provided by SBO.  GBT88 |
| username | mandatory | string(20) | opcode + player Id |
| amount | mandatory | string | decimal value: up to 2 decimal points.  Eg. 12.21, 9, full  Only positive number is allowed.  If pass value as string “full” in the field, all available balance will be withdrawn. Otherwise specified the positive number to withdraw,  Unrecognized money format or not “full” will be treated as 0. |
| txnId | mandatory | string(19) | It must be unique among all transactions (deposit/withdraw) in the IMS.  All txnId is prepend with opcode. |
| access | mandatory | string(32) | MD5 Hashing (take the path from after the ‘sportsfundservice/’ + ‘/’ + Private Key |

Example

|  |
| --- |
| http://103.9.150.43/sportsfundservice/withdrawal/M777/player/aaaa32/amount/full/txnid/aaaa32-18?access=84550f3e6ccd37f1de66287934b0ded7 |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error | mandatory | Error | error |
| serverId | mandatory | string | server id |
| txnId | mandatory | string(30) | prepend with opcode and “-“ |
| refno | mandatory | string(64) | generated by SBO system and unique among all transactions. |
| amount | mandatory | decimal | return transaction amount if no error.  redturn 0 when any error. |
| balance | mandatory | decimal | player Gsp Balance |
| outstanding | mandatory | decimal | ?? |

Error

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| id | mandatory | string | error id |
| msg | mandatory | string | error message |

Example

|  |
| --- |
| {  “txnId” : “M777-14-218-222”,  “refno” : “WW1-8217-WZZ218”,  “amount” : 150.00,  “balance” : 0.0,  “outstanding” : 0.0,  “error” : { “id” : 0, “msg” : “no error”},  “serverId” : “RSP-07”  } |

* + 1. CheckTransactionStatus (GET)

**Mount Point:** http://103.9.150.43/sportsfundservice/checktransactionstatus/{**opcode**}/txnid/{**txnId**}?access={**access**}

This API is to check the transaction status by providing transaction id.

#### Request

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| opcode | mandatory | string(10) | The operator code provided by SBO.  GBT88 |
| txnId | mandatory | string(19) | It must be unique among all transactions (deposit/withdraw) in the IMS.  All txnId is prepend with opcode. |
| access | mandatory | string(32) | MD5 Hashing (take the path from after the ‘sportsfundservice/’ + ‘/’ + Private Key |

Example

|  |
| --- |
| http://103.9.150.43/sportsfundservice/checktransactionstatus/M777/txnid/aaaa32-18?access=189b3a0c90bbc872abfdaf0a26cac5ca |

#### Response

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error | mandatory | Error | error |
| serverId | mandatory | string | server id |
| txnId | mandatory | string(30) | prepend with opcode and “-“ |
| refno | optional | string(64) | If the transaction is successful, a valid refno will return. Otherwise, it will be empty. |
| transactionType | mandatory | string(15) | Deposit or Withdraw |
| amount | mandatory | decimal | return transaction amount if no error.  redturn 0 when any error. |

Error

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| id | mandatory | string | error id |
| msg | mandatory | string | error message |

Example

|  |
| --- |
| {  “txnId” : “M777-14”,  “refno” : “X1-8217-DZZ00218”,  “transactionType” : “withdraw”,  “amount” : 1312.93,  “error” : { “id” : 0, “msg” : “no error”},  “serverId” : “RSP-07”  } |

* + 1. GetPlayerBalance (GET)

**Mount Point:** http://103.9.150.43/sportsfundservice/getplayerbalance/{**opcode**}/player/{**username**}?access={**access**}

This API is to retrieve a player’s balance from SBO system.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| opcode | mandatory | string(10) | The operator code provided by SBO.  GBT88 |
| username | mandatory | string(20) | opcode + player Id |
| access | mandatory | string(32) | MD5 Hashing (take the path from after the ‘sportsfundservice/’ + ‘/’ + Private Key |

Example

|  |
| --- |
| http://103.9.150.43/sportsfundservice/getplayerbalance/M777/player/aaaa32?access=133aef39e784ef91142879d115b134220 |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error | mandatory | Error | error |
| serverId | mandatory | string | server id |
| username | mandatory | string(30) | opcode + player id |
| currency | mandatory | string | Currency is based on ISO 4217 standard.  Only THB, IDR, VND, KRW, JPY are Country’s IP restricted.  SGD Singapore Dollar (Base Currency)  MYR Malaysia Ringgit  CNY China Yuan Renminbi  THB Thailand Baht  KRW South Korean Won  IDR Indonesia Rupiah  VND Vietnam Dong  HKD Hong Kong Dollar  JPY Japan Yen |
| balance | mandatory | decimal | decimal value: up to 2 decimal points. |
| outstanding | mandatory | decimal | the wager pending for settle |

Error

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| id | mandatory | string | error id |
| msg | mandatory | string | error message |

Example

|  |
| --- |
| {  “username” : “aaa01”,  “currency” : “SGD”,  “Balance” : 121.23,  “outstanding” : 22.00,  “error” : { “id” : 0, “msg” : “no error”},  “serverId” : “RSP-07”  } |

* + 1. Logout (GET)

**Mount Point:** http://103.9.150.43/sportsfundservice/logout/{**opcode**}/player/{**username**}?access={**access**}

This API is to logout player and clear the current session if the player is in the SBO system.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| opcode | mandatory | string(10) | The operator code provided by SBO.  GBT88 |
| username | mandatory | string(20) | opcode + player Id |
| access | mandatory | string(32) | MD5 Hashing (take the path from after the ‘sportsfundservice/’ + ‘/’ + Private Key |

Example

|  |
| --- |
| http://103.9.150.43/sportsfundservice/logout/M777/player/aaaa32?access=93a8325ac75bf430d68a6ef31b7a180b |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error | mandatory | Error | error |
| serverId | mandatory | string | server id |

Error

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| id | mandatory | string | error id |
| msg | mandatory | string | error message |

Example

|  |
| --- |
| {  “error” : { “id” : 0, “msg” : “no error”},  “serverId” : “RSP-07”  } |

* + 1. Ping (GET)

**Mount Point:** http://103.9.150.43/sportsfundservice/ping/{**opcode**}?access={**access**}

This API is to make an application level ping to make sure the API server available.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| opcode | mandatory | string(10) | The operator code provided by SBO.  GBT88 |
| access | mandatory | string(32) | MD5 Hashing (take the path from after the ‘sportsfundservice/’ + ‘/’ + Private Key |

Example

|  |
| --- |
| http://103.9.150.43/sportsfundservice/ping/M777?access=E17cd4adf393326d80a56375383d8f0f |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error | mandatory | Error | error |
| serverId | mandatory | string | server id |

Error

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| id | mandatory | string | error id |
| msg | mandatory | string | error message |

Example

|  |
| --- |
| {  “error” : { “id” : 0, “msg” : “no error”},  “serverId” : “RSP-07”  } |

* + 1. CheckStatus (GET)

**Mount Point:** http://103.9.150.43/sportsfundservice/checkstatus/{**opcode**}?access={**access**}

This API is to to deliver the maintenance information.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| opcode | mandatory | string(10) | The operator code provided by SBO.  GBT88 |
| access | mandatory | string(32) | MD5 Hashing (take the path from after the ‘sportsfundservice/’ + ‘/’ + Private Key |

Example

|  |
| --- |
| http://103.9.150.43/sportsfundservice/checkstatus/M777?access=3c17ce119d45994c4db7641ce91a831a |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error | mandatory | Error | error |
| serverId | mandatory | string | server id |
| isUM | mandatory | bool | true: currently on is under maintenance  false: otherwise |
| umStartTime | optional | string | Available if UM period is not past |
| umEndTime | optional | string | Available if UM period is not past |

Error

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| id | mandatory | string | error id |
| msg | mandatory | string | error message |

Example

|  |
| --- |
| {  “isUM” : true,  “umStartTime” : “2014-10-10T10:30:00-04:00”,  “umEndTime” : “2014-10-10T11:00:00-04:00”,  “error” : { “id” : 0, “msg” : “no error”},  “serverId” : “RSP-07”  } |

* + 1. GetCustomerReport (GET)

**Mount Point:** http://103.9.150.43/sportsfundservice/getcustomerreport/{**opcode**}/user/{**username**}/type/{**type**}?startdate={**startDate**}&enddate={**endDate**}&access={**access**}

If the type is agent, then return all player’s turnover report that under this agent.

If the type is player, then return the specific player’s turnover report.

All reports are only showing today and the past 89 days data (total 90 days).

The date is GMT -4. So, you pass the date must be GMT offset value cause the startDate and endDate in the system follow GMT -4.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| opcode | mandatory | string(10) | The operator code provided by SBO.  GBT88 |
| username | mandatory | string(20) | opcode + player id |
| type | mandatory | string(15) | agent or player |
| startDate | mandatory | string | YYYY-MM-DD |
| endDate | mandatory | string | YYYY-MM-DD |
| access | mandatory | string(32) | MD5 Hashing (take the path from after the ‘sportsfundservice/’ + ‘/’ + Private Key |

Example

|  |
| --- |
| <http://103.9.150.43/sportsfundservice/getcustomerreport/M777/user/aaa08/type/agent?startdate=2013-10-12&enddate=2014-10-10&access=0dae956579023f0cca04bd3a246b640e>  <http://103.9.150.43/sportsfundservice/getcustomerreport/M777/user/aaaa22/type/player?startdate=2013-10-12&enddate=2014-10-10&access=c452c427008379a0355711c90933c6d8> |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error | mandatory | Error | error |
| serverId | mandatory | string | server id |
| playerRevenue | mandatory | PlayerRevenue | If type is agent, return all player’s turnover report under this agent, or return the player’s turnover report |

Error

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| id | mandatory | string | error id |
| msg | mandatory | string | error message |

PlayerRevenue

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| username | mandatory | string(20) | opcode + playerId |
| betcount | mandatory | BetCount | The bet count of the player between start date and end date, include all status of bet. |
| turnover | mandatory | Turnover | The Turnover of the player between start date and end date, include all status of bet. |
| winLose | mandatory | decimal | decimal value: up to 2 decimal points.  The total winlose of the player between start date and end date |
| commission | mandatory | decimal | decimal value: up to 2 decimal points.  The total commission of the player between start date and end date |
| totalDeposit | mandatory | decimal | decimal value: up to 2 decimal points.  The total deposits of the player between start date and end date |
| totalWithdrawal | mandatory | decimal | decimal value: up to 2 decimal points.  The total withdrawals of the player between start date and end date |

BetCount

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| total | mandatory | int | the total bet count |
| waiting | mandatory | int | the waiting bet count |
| running | mandatory | int | the running bet count |
| lose | mandatory | int | the lose bet count |
| won | mandatory | int | the won bet count |
| draw | mandatory | int | the draw bet count |
| void | mandatory | int | the voided bet count |
| reject | mandatory | int | the rejected bet count |
| refund | mandatory | int | the refunded bet count |

Turnover

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| total | mandatory | decimal | the total bet turnover |
| waiting | mandatory | decimal | the waiting bet turnover |
| running | mandatory | decimal | the running bet turnover |
| lose | mandatory | decimal | the lose bet turnover |
| won | mandatory | decimal | the won bet turnover |
| draw | mandatory | decimal | the draw bet turnover |
| void | mandatory | decimal | the voided bet turnover |
| reject | mandatory | decimal | the rejected bet turnover |
| refund | mandatory | decimal | the refunded bet turnover |

Example

|  |
| --- |
| {  “playerRevenue” : [{  “username” : “ABCplayer01”,  “betcount” : {  “total”: 70,  “waiting”:5,  “running”:5,  “lose”:25,  “won”:25,  “draw”:5,  “void”:0,  “reject”:5,  “refund”:0  },  “turnover” :{  “total”: 700.00,  “running”:50.00,  “lose”:250.00,  “won”:250.00,  “draw”:50.00,  “void”:0.00,  “reject”:50.00,  “refund”:0.00,  },  “Winlose” : 100,  “commission”: 10,  “TotalDeposit” : 0.00,  “TotalWithdrawal”: 0.00,  }],  “error” : { “id” : 0, “msg” : “no error”},  “serverId” : “RSP-07”  } |

* + 1. GetCustomerBetList(GET)

**Mount Point:** http://103.9.150.43/sportsfundservice/getcustomerbetlist/{**opcode**}/user/{**username**}?startdate={**startDate**}&enddate={**endDate**}&access={**access**}

If the type is agent, then return all player’s turnover report that under this agent.

If the type is player, then return the specific player’s turnover report.

All reports are only showing today and the past 89 days data (total 90 days).

The date is GMT -4. So, you pass the date must be GMT offset value cause the startDate and endDate in the system follow GMT -4.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| opcode | mandatory | string(10) | The operator code provided by SBO.  GBT88 |
| username | mandatory | string(20) | opcode + player id |
| startDate | mandatory | string | YYYY-MM-DD |
| endDate | mandatory | string | YYYY-MM-DD |
| access | mandatory | string(32) | MD5 Hashing (take the path from after the ‘sportsfundservice/’ + ‘/’ + Private Key |

Example

|  |
| --- |
| http://103.9.150.43/sportsfundservice/getcustomerbetlist/M777/user/aaaa22?startdate=2013-10-12&enddate=2014-10-10&access=c452c427008379a0355711c90933c6d8 |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error | mandatory | Error | error |
| serverId | mandatory | string | server id |
| playerBetList | mandatory | List<BetDetail> | the list of bet under the agent of player |
| turnover | mandatory | decimal | the turnover of the player bet list |

Error

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| id | mandatory | string | error id |
| msg | mandatory | string | error message |

BetDetail

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| refno | mandatory | string(50) | reference number for bet |
| username | mandatory | string(20) | opcode + playerId |
| sportType | mandatory | string(50) | sport type of bet |
| orderTime | mandatory | string | time that player places a bet.  yyyy-MM-dd hh:mm:ss TT”.  Time zone is GMT-4. |
| odds | mandatory | decimal | the odds player places |
| oddsStyle | mandatory | string(1) | ‘H’ for HK odds, ’M’ for Malay odds, ‘E’ for Euro odds, ’I’ for Indo |
| stake | mandatory | decimal | the stake player places |
| actualStake | mandatory | decimal | decimal value: up to 2 decimal points.  The total winlose of the player between start date and end date |
| currency | mandatory | string(3) | the currency player uses |
| status | mandatory | string(10) | The field Indicates the bet status, possible values are “waiting”, “running”, ”won”, ”lose”, ”draw”, ”waiting rejected”, ”refunded”, “voided” |
| winlose | mandatory | decimal | 0 mean the player lost his all actual stake.  when winlose= actualstake, the result is draw. winlose>actualstake, the result is won by player. |
| turnover | mandatory | decimal | decimal value: up to 2 decimal points.  The amount contributing to turnover.  The amount should be the same unless the bet is voided or refunded. |
| isLive | mandatory | bool | the bet is live or not |
| subBet | mandatory | List<Subbet> | The field contain the bet detail.  The field contains more than 2 item only when bet is mix parley. |

Subbet

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| betOption | mandatory | string(10) | the bet option player chooses |
| marketType | mandatory | string(10) | the market type of bet |
| hdp | mandatory | decimal | the number indicates the handicap number or OU number |
| isLive | mandatory | bool | the bet is live or not |
| odds | mandatory | decimal | the odds player place |
| league | mandatory | string(50) | the league of bet |
| match | mandatory | string(100) | The format is “Home team vs Away Team” |
| winLoseDate | mandatory | string | yyyy-MM-dd 00:00:00  The date indicates the date the event takes place |
| liveScore | mandatory | string | The score of live.  It’s “0:0” when the bet is non-live. |
| customizedBetType | optional | string | Description for customized bet type. The field is proved when the bet is customized type, ex: number of red card. |
| betCondition | optional | string | Only available when bet is placed at AOS of live correct score. |

Example

|  |
| --- |
| {  ‘playerBetList’ : {  username:"ABCtesewuf",  actualStake:80,  currency:"THB",  odds:-0.8,  oddsStyle:"M",  orderTime:"2015-02-04 10:09:40",  refNo:"221929192i",  sportType:"Football",  stake:100,  status:"running",  subBet:[  {  betType:"H",  hdp:"0",  league:"Turkey Cup",  liveScore:"0:0",  marketType:"HDP",  match:"Altinordu (n) vs Kayserispor",  odds:-0.8,  winLoseDate:"2015-02-04 00:00:00"  }  ]  },  turnover:80 ,  isLive:"false",  ‘error’ : { ‘id’ : 0, ‘msg’ : ‘no error’},  ‘serverId’ : ‘RSP-07’  } |

* 1. IBC (IBC)
     1. CreateMember (GET)

**Mount Point:** http://api.mid.ib.gsoft88.net/api/CreateMember?OpCode={**Opcode**}&PlayerName={**PlayerName**}&FirstName={**FirstName**}&LastName={**LastName**}&OddsType={**OddsType**}&Currency={**Currency**}&MaxTransfer={**MaxTransfer**}&Min

Transfer={**MinTransfer**}&SecurityToken={**SecurityToken**}

The API is used to create new member account on IBC Game Service Provider.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Opcode | mandatory | string(10) | The operator code provided by IBC. |
| PlayerName | mandatory | string(20) | player Id |
| SecurityToken | mandatory | string(50) | MD5 Hashing  private key + “/”+ string begin from “api” until the end of string. |
| OddsType | mandatory | int | 1 Malay Odds  2 Hong Kong Odds  3 Decimal Odds  4 Indo Odds  5 American Odds |
| Currency | mandatory | int | **1** SGD Singapore Dollar  **2** MYR Malaysian Ringgit  **3** USD US Dollar  **4** THB Thai Baht  **5** HKD Hong Kong Dollar  **6** EUR Euro  **9** AUD Australian Dollar  **12** GBP British Pound  **13** RMB Chinese Renminbi  **15** IDR Indonesia Rupiah  **20** UUS Virtual Dollar (for testing propose)  **32** JAP Japan Yen  **41** CHF Swiss Franc  **45** WON Korean Won  **46** BND Brunei Dollar  **49** MXN Mexican Peso  **50** CAN Canadian Dollar  **51** VND Vietnamese Dong  **52** DKK Danish Krone  **53** SEK Swedish Krona  **54** NOK Norwegian Krone  **55** RUB Russian Ruble  **56** PLN Polish Zloty  **57** CZK Czech Koruny  **58** RON Romanian Leu  **61** INR Indian Rupee |
| MaxTransfer | mandatory | Decimal(18,3) | Max Transfer between Sportsbook System and Licensee  System |
| MinTransfer | mandatory | Decimal(18,3) | Min Transfer between Sportsbook System and Licensee  System |
| FirstName | optional | string(50) | first name of player |
| LastName | optioanl | string(32) | last name of player |

Example

|  |
| --- |
| http://api.mid.ib.gsoft88.net/api/CreateMember?OpCode=local&PlayerName=playe  r1&FirstName=Player&LastName=One&OddsType=1&Currency=20&MaxTransfer=1000&Min  Transfer=50&SecurityToken=063814D2734CF00D78E5F7862ACFC8C3 |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error\_code | mandatory | int | error code |
| message | mandatory | string | error message |

Example

|  |
| --- |
| {  "error\_code": 0,  "message": "Successfully executed"  } |

* + 1. Login (GET)

**Mount Point:** http://api.mid.ib.gsoft88.net/api/Login?OpCode={**Opcode**}&PlayerName={**PlayerName**

&SecurityToken={**SecurityToken**}

The API is used to login the IBC Game Service Provider.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Opcode | mandatory | string(10) | The operator code provided by IBC. |
| PlayerName | mandatory | string(20) | player Id |
| SecurityToken | mandatory | string(50) | MD5 Hashing  private key + “/”+ string begin from “api” until the end of string. |

Example

|  |
| --- |
| http://api.mid.ib.gsoft88.net/api/Login?OpCode=local&PlayerName=player1&Secu  rityToken=6AFC7E3AD8F7F61285B452B41F613A24 |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error\_code | mandatory | int | error code |
| message | mandatory | string | error message |
| sessionToken | mandatory | string | session token used when launch the game url |

Example

|  |
| --- |
| {  "error\_code": 0,  "message": "Successfully executed"  } |

* + 1. FundTransfer (GET)

**Mount Point:** http://api.mid.ib.gsoft88.net/api/FundTransfer?OpCode=local&PlayerName=player1&OpTransId=A00000001&Amount=100&Direction=1&SecurityToken=B0EC8069013C4B3526A39B9B4C8982A9

The API is used to login the IBC Game Service Provider.

If response’s status is “Pending”(2), should call CheckFundTransfer and confirm the result.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Opcode | mandatory | string(10) | The operator code provided by IBC. |
| PlayerName | mandatory | string(20) | player Id |
| SecurityToken | mandatory | string(50) | MD5 Hashing  private key + “/”+ string begin from “api” until the end of string. |
| OpTransId | mandatory | string(50) | transaction id |
| amount | mandatory | decimal(18,3) | transfer amount |
| direction | mandatory | int | 0: withdraw (IBC to IMS)  1: deposit (IMS to IBC) |

Example

|  |
| --- |
| http://api.mid.ib.gsoft88.net/api/FundTransfer?OpCode=local&PlayerName=player1&OpTransId=A00000001&Amount=100&Direction=1&SecurityToken=B0EC8069013C4B3526A39B9B4C8982A9 |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error\_code | mandatory | int | error code |
| message | mandatory | string | error message |
| Data | mandatory | Data | transfer result |

Data

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| trans\_id | mandatory | long | IBC transaction id |
| before\_amount | mandatory | decimal | member balance before the fund transfer |
| after\_amount | mandatory | decimal | member balance after the fund transfer |
| status | mandatory | int | 0 OK Successful executed  1 Failed System Error  2 Pending Status Unknown, call **CheckFundTransfer** Method to confirm |

Example

|  |
| --- |
| {  "Data": {  "trans\_id": 81,  "before\_amount": 0,  "after\_amount": 100,  "status": 0,  },  "error\_code": 0,  "message": "Success"  } |

* + 1. CheckUserBalance (GET)

**Mount Point:** http://api.mid.ib.gsoft88.net/api/CheckUserBalance?OpCode={**Opcode**}&PlayerName={**PlayerName**}

&SecurityToken={**SecurityToken**}

The API is used to login the IBC Game Service Provider.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Opcode | mandatory | string(10) | The operator code provided by IBC. |
| PlayerName | mandatory | string(20) | player Id |
| SecurityToken | mandatory | string(50) | MD5 Hashing  private key + “/”+ string begin from “api” until the end of string. |

Example

|  |
| --- |
| http://api.mid.ib.gsoft88.net/api/CheckUserBalance?OpCode=local&PlayerName=p  layer1&SecurityToken=244A3FA78FF20EF87D64AAD2363DB711 |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error\_code | mandatory | int | error code |
| message | mandatory | string | error message |
| Data | mandatory | Data | transfer result |

Data

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| playerName | mandatory | long | IBC player name |
| balance | mandatory | decimal | IBC player balance amount |
| outstanding | mandatory | decimal | IBC player outstanding amount |
| currency | mandatory | int | IBC currency number  should be needed to parse IMS currency system |

Example

|  |
| --- |
| {  "Data": [  {  "playerName": "player1",  "balance": 0,  "outstanding": 0,  "currency": 20,  }  ],  "error\_code": 0,  "message": "Successfully executed"  } |

* + 1. CheckFundTransfer (GET)

**Mount Point:** http://api.mid.ib.gsoft88.net/api/CheckFundTransfer?OpCode={**Opcode**}&PlayerName={**PlayerName**}

&OpTransId={**OpTransId**}&SecurityToken={**SecurityToken**}

The API is used to login the IBC Game Service Provider.

***Request***

Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| Opcode | mandatory | string(10) | The operator code provided by IBC. |
| PlayerName | mandatory | string(20) | player Id |
| SecurityToken | mandatory | string(50) | MD5 Hashing  private key + “/”+ string begin from “api” until the end of string. |
| OpTransId | mandatory | string(50) | IMS transaction id |

Example

|  |
| --- |
| http://api.mid.ib.gsoft88.net/api/CheckFundTransfer?OpCode=local&PlayerName=  player1&OpTransId=A00000001&SecurityToken=318BFBE347F467362F5D46364251C95C |

***Response***

Response Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| error\_code | mandatory | int | error code |
| message | mandatory | string | error message |
| Data | mandatory | Data | transfer result |

Data

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **MANDATORY** | **DATA TYPE** | **DESCRIPTION** |
| trans\_id | mandatory | long | IBC transaction id |
| before\_amount | mandatory | decimal | member balance before the fund transfer |
| after\_amount | mandatory | decimal | member balance after the fund transfer |
| status | mandatory | int | 0 OK Successful executed  1 Failed System Error  2 Pending Status Unknown, call **CheckFundTransfer** Method to confirm |

Example

|  |
| --- |
| {  "Data": [  {  "playerName": "player1",  "balance": 0,  "outstanding": 0,  "currency": 20,  }  ],  "error\_code": 0,  "message": "Successfully executed"  } |

# **Platform**

# **Appendix**

* 1. Terms Clarification
     1. API

|  |  |
| --- | --- |
| **TERM** | **DESCRIPTION** |
| BO API | API between BO and IMS |
| Front API | API between Front web and IMS |
| GSP API | api between GSP and IMS |

* + 1. Business Terms

|  |  |
| --- | --- |
| **TERM** | **DESCRIPTION** |
| Agent |  |
| Affiliate |  |
| Admin |  |
| Player |  |
| Member |  |

* + 1. Catogory ID Description

|  |  |  |
| --- | --- | --- |
| **ID** | **PASSWORD** | **DESCRIPTION** |
| gspId | gspPassword | Game service provider + Id/Password  Given from GSP.  This is a general id/password to be able to access GSP API. |
| agentId | agentPassword | Agent Id/Password  Given Id/Password from Back office administrator  This is used to check whether the this agent is rightful or not |
| adminId | adminPassword | Id/Password to be used when the agent admin login the back office. |
| nickname | playerPassword | nickname /Password to be used when the player login the front web. |
| playerId | N/A | auto-generated sequence id according to signup order of specific agent |
| affAdminId | AffiliateAdminPassword | Id/Password to be used when the affiliate admin login the back office |
| gspMemberId | gspMemberPassword | Id/Password to be used when the agent player play game in specific GSP. |
| gspExid | N/A | GSP external id which is generated from gsp DB. |
| gspGameCurrencyId | gspGameCurrencyPassword | Some specific game have different gspIds regarding to currencies. (XPro, XTD)  if gspGameCurrencyId exsits, should use this one instead of gspId when invoking GSP API  This id is same as branch id in source code.  refer to database (tb\_cType)  Game service provider Currency + Id/Password. |

* + 1. agAccountTransaction vs gspBetTransaction

|  |  |
| --- | --- |
| **TERM** | **DESCRIPTION** |
| agAccountTransaction | Transaction to change both agent account balance and gspMember balance  In case of transfer deposit, a gspMemberBalance is increased as much as amount of transaction. An opBalance is decreased.  In case of transfer withdraw, it works vice versa. |
| gspBetTransaction | Transaction to change only gspMember balance  it is to be called a bet transaction whenever a gspMember play game (place a bet or win the game and so on) |

* + 1. Wallet Clarification

|  |  |
| --- | --- |
| **TERM** | **DESCRIPTION** |
| Single Wallet | a player’s balance in IMS  a.k.a seamless wallet, main wallet |
| Wallet to Wallet | a player’s balance in GSP  In case of WalletToWallet, IMS is only responsible for transferring between IMS and GSP. IMS can only know total amount of GSP balance, but can’t know the history of GSP Balance except money transferring. By calling History API, IMS get the detailed information of Balance change regarding to the bet. |

* + 1. Sports Bet Type (WFT)

|  |  |
| --- | --- |
| **TERM** | **DESCRIPTION** |
| HDP |  |
| OU |  |
| OE |  |
| 1X2 | 1: Home X: draw 2: Away |
| PAR | parlay: 조합 |
| ORT | outright :승자 예상 베팅 |
| CS | Correct Score |
| TG |  |
| HFT |  |
| FLG |  |
| A |  |
| B |  |
| C |  |
| D |  |
| E |  |
| F |  |
| 4 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Statement | | |
| Share | | Comm |
| Live=0 | Live=1 |  |
| Full Handicap | Non live HDP/OU/OE | Live HDP/OU/OE | Group |
| First Handicap | Non live HDP/OU/OE | Live HDP/OU/OE | Group |
| Over/Under | Non live HDP/OU/OE | Live HDP/OU/OE | Group |
| First Over/Under | Non live HDP/OU/OE | Live HDP/OU/OE | Group |
| 1X2 | Non live 1X2 | Live 1X2 | 1X2 |
| First 1X2 | Non live 1X2 | Live 1X2 | 1X2 |
| Odd/Even | Non live HDP/OU/OE | Live HDP/OU/OE | Group |
| First Odd/Even | Non live HDP/OU/OE | Live HDP/OU/OE | Group |
| Correct Score | Others | Others | Others |
| First Correct Score | Others | Others | Others |
| Total Goal | Others | Others | Others |
| First Goal/Last Goal | Non live HDP/OU/OE | Non live HDP/OU/OE | Others |
| Half Time/Full Time | Others | Others | Others |
| ML | Others | Others | Group |
| Outright | Others | Others | Others |
| Mix Parlay | MixParlay | MixParlay | MixParlay |

* + 1. Sports Bet Limit (ASC)

|  |  |
| --- | --- |
| **TERM** | **DESCRIPTION** |
| MinBet |  |
| MaxBet |  |
| PerMaxBet |  |
| MixMinBet |  |
| MixMaxBet |  |
| DayLimit |  |

* 1. Transaction ID Management

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Wallet** | **GAME SERVICE**  **PROVIDER** | **TRANSACTION ID**  **EXIST** | **TRANSACTION ID**  **SUBSTIBUTE** | **DESCRIPTION** |
| Single | W88 | true |  | trx\_id: Unique transaction ID from Hudb |
|  | GD | false | message id | “D”+YYMMDDhhmmss + 5 random char  “W”+YYMMDDhhmmss + 5 random char |
|  | Micro | true | transaction id | IMS transaction id |
|  | EZ | true |  | lpsTxId and |
| WalletToWallet | WSC | true |  | serial number |
|  | AG | false | billno | cagent + sequence(13~16 digit, unique) |
|  | AM |  |  |  |
|  | AS | true |  | serial number |
|  | Moon | false | unique\_code | unique code from MemberLogin().  Should get new unique code before calling transfer.  What happen if request with same unique\_code again? |
|  | XPro | true |  | transactionID |
|  | XTD | true |  | tranRefID |
|  | PT | true |  | externalTranId |
|  | PG |  |  |  |
|  | TTGS |  |  |  |
|  | WSC | true |  | serial number |

* 1. Language Code Management

http://en.wikipedia.org/wiki/List\_of\_ISO\_639-2\_codes

|  |  |
| --- | --- |
| **DISPLAY TYPE** | **EXAMPLE** |
| ISO 639-1 | two-letter codes, one per language or ISO 639 macrolanguage  en |
| ISO 639-2 | ZHO or CHI |
| ISO 639-3 | ZHO |
| Web Browser Language Identification Codes (HTTP\_ACCEPT\_LANGUAGE) | zh  zh-CN: Chinese  zh-HK: Chinese(Hongkong) |

* 1. Currency Code Management

http://en.wikipedia.org/wiki/ISO\_4217

|  |  |
| --- | --- |
| **DISPLAY TYPE** | **EXAMPLE** |
| ISO 4217 | USD  EUR |

* 1. Country Code Management

https://www.iso.org/obp/ui/#search

|  |  |
| --- | --- |
| **DISPLAY TYPE** | **EXAMPLE** |
| ISO 3166 alpha-2 | KR |
| ISO 3166 alpha-3 | KOR |

**­**