Version 2017-11-01

>> Fetch Ticket Info add Currency Code

Version 2017-11-06

>> 2.2 Update comtype added V

Version 2017-11-14

>> Fetch ticket info CS > CSR,

Double Chance

<info> X2, 1X, 12 (X=Draw, 1=Home, 2=Away)

Version 2017-12-12

>> Fetch Ticket Info add in Transtype PAS (STEP), STEP ticket Details

Version 2017-12-15

>> 2.2 Update added PAS (STEP) Setting fields

Sportsbook API

Introduction

1. Request Format

http://<host>/api.aspx?secret=<secret>&action=<action>&agent=<agent>&username=<username>&...

1.1 Security

a) Secret

1.2 Response Format

```
<?xml version="1.0"?>

<response>
        <errcode><%=ErrCode %></errcode>
        <errtext><%=ErrText %></errtext>
        <result><%=Result %></result>
</response>
```

1.3 Error

ErrCode = 0 is no error.

Negative ErrCode value is general error.

Positive ErrCode value is action specific error.

```
// Maintenance
public const int CODE_MAINTENANCE = -1;
public const string TEXT_MAINTENANCE = "System maintenance";
// Invalid secret
public const int CODE_INVALID_SECRET = -2;
public const string TEXT_INVALID_SECRET = "Invalid secret";
// IP address not allow
public const int CODE_IP_NOT_ALLOW = -3;
public const string TEXT_IP_NOT_ALLOW = "Ip address not allow";
// Invalid username
public const int CODE_INVALID_USERNAME = -4;
public const string TEXT_INVALID_USERNAME = "Invalid username (length 1-16, format A-Za-z0-9)";
// Invalid action
public const int CODE_INVALID_ACTION = -5;
public const string TEXT_INVALID_ACTION = "Invalid action";
// Invalid agent CompType
public const int CODE_INVALID_AGENT_COMPTYPE = -7;
public const string TEXT_INVALID_AGENT_COMPTYPE = "Invalid agent";
// Invalid username CompType
public const int CODE_INVALID_USERNAME_COMPTYPE = -8;
public const string TEXT_INVALID_USERNAME_COMPTYPE = "Invalid username";
// Invalid input
public const int CODE_INVALID_INPUT = -100;
// Exception
public const int CODE_EXCEPTION = -101;
```

2. Actions

2.1 Create Member

Request:

action=create

Example:

http://<host>/api.aspx?secret=<secret>&action=create&agent=<agent>&username=<username>

Response:

```
// Errors:
//          1, Username exist
//          3, <error>
// Result:
//          username
```

2.2 Update

Update member max bets, limits and commissions settings.

Request:

action=update

max1=<Max Bet HDP/OU/OE>

max2=<Max Single Bet Limit Parlay>

max3=<Max Bet Correct Score>

max4=<Max Bet Step>

lim1=<Max Per Match HDP/OU/OE >

lim2=<Mix Parlay Limit Per Combination>

lim3=<Max Per Match Correct Score>

lim4=<Step Limit Per Combination>

comtype=<Choice of A,B,C,D,E,F,4,V for HDP/OU/OE>

com1=<Commission for HDP/OU/OE>

com2=<Commisson for 1X2/Outright>

com3=<Commission for PAR/CS/TG>

com4=<Commission for Step (2 match)>

com5=<Commission for Step (3-4 match)>

com6=<Commission for Step (5-6 match)>

com7=<Commission for Step (7-8 match)>

com8=<Commission for Step (9-10 match)>

com9=<Commission for Step (11-12 match)>

suspend=<0: no suspend, 1:suspend>

Example:

http://<host>/api.aspx?secret=<secret>&action=update&agent=<agent>&username=<username>&max1=1100&max2=1200&max3=1300&max4=1400&lim1=1000&lim2=1100&lim3=1200&lim4=1300&comtype=A&com1=0.1&com2=0.2&com3=0.3&com4=0.4&com5=0.5&com6=0.6&com7=0.7&com8=0.8&com9=0.0&suspend=0

Response:

```
// Errors:
//          1, <error>
// Result:
```

2.3 Check Balance

Request:

action=balance

Example:

http:// <host>/api.aspx?secret=<secret>&agent=<agent>&username=<username>&action=balance

Response:

```
// Errors:
//          1, User not exist
//          3, System maintainenace
//          4, Casino online
//          5, <error>
// Result:
//          balance
```

2.4 Deposit

You must call "Check Balance" with no error before deposit.

Request:

action=deposit

amount=<amount>

serial=<serial number>

Example:

http://<host>/api.aspx?secret=<secret>&agent=<agent>&username=<username>&action=deposit&serial=001&amount=100

Response:

```
// Errors:
//          1, <error>
// Result:
//          balance
```

2.5 Withdraw

You must call "Check Balance" with no error before withdraw.

Request:

action=withdraw

amount=<amount>

serial=<serial number>

Example:

http://<host>/api.aspx?secret=<secret>&agent=<agent>&username=<username>&action=withdraw&serial=002&amount=50

Response:

```
// Errors:
//          1, <error>
// Result:
//          balance
```

2.6 Login

Request:

action=login

host=<betting page domain>

lang=<language (EN-US, ZH-CN, KO-KR, TH-TH, VI-VN, ES-PA)>

accType=<odds Type (MY, HK, ID, EU)>

Example:

http://<host>/api.aspx?secret=<secret>&agent=<agent>&username=<username>&action=login&host=<betting host>&lang=EN-US&accType=HK

Response:

```
// Error:
//          1, Internal error: invalid LoginPrivateKey
// Result:
            <login>
                 <host>[betting host]</host>
                 <param>
                        <us>[value of us]</us>
                        <k>[value of k]</k>
                        <lang>[value of lang]</lang>
                        <accType>[value of accType]</accType>
                        <r>[value of r]</r>
                 </param>
            </login>
```

In order to login you need build the login URL yourself.

Example:

http://<host>?us=<us>&k=<k>&lang=<lang>&accType=<accType>&r=r

2.7 Logout

Request:

action=logout

Example:

http://<host>/api.aspx?secret=<secret>&agent=<agent>&username=<username>&action=logout

Response:

```
// No error
```

2.8 Check Payment

Request:

action=check_payment

serial=<serial>

If username is empty, check all users.

Example:

http://<host>/api.aspx?secret=<secret>&agent=<agent>&username=<username>&action=check_payment&serial=001

Response:

```
// Errors:
//          1, <error>
// Result:
//          List of payments

public class Payment
{
            public string Username;
            public double Amount;
}
```

2.9 Check Online Users

Request:

action=check_online

If username is empty, check all users.

Example:

http://<host>/api.aspx?secret=<secret>&agent=<agent>&username=<username>&action=check_online

Response:

```
// Errors:
//          1, <error>
// Result:
//          List of usernames
//          Example: john,lenny,peter
```

2.10 Parlay

Request:

action=parlay

ticket_id=<ticket_id>

Example:

http://<host>/api.aspx?secret=<secret>&agent=<agent>&username=<username>&action=parlay&ticket_id=PAR753983515

Response:

List of tickets.

2.101 Step

Request:

action=step

ticket_id=<ticket_id>

Example:

http://<host>/api.aspx?secret=<secret>&agent=<agent>&username=<username>&action=step&ticket_id=

PAS753983515

Response:

List of tickets.

2.11 Team

Request:

action=team

team_id=<team_id>

Example:

http://<host>/api.aspx?secret=<secret>&agent=<agent>&username=<username>&action=team&team_id=1024

Response:

<name><lang>en-US</lang><txt>Arsenal</txt></name>...

2.12 League

Request:

action=league

league_id=<league_id>

Example:

http://<host>/api.aspx?secret=<secret>&agent=<agent>&username=<username>&action=league&league_id=500

Response:

<name><lang>en-US</lang><txt>UEFA</txt></name>...

2.13 SportsType

Request:

action=sportstype

sportstype _id=< sportstype _id>

Example:

http://<host>/api.aspx?secret=<secret>&agent=<agent>&username=<username>&action= sportstype & sportstype _id=7

Response:

<name><lang>en-US</lang><txt>Basketball</txt></name>...

2.14 Match Count

Request:

action=match_count

Example:

http://<host>/api.aspx?secret=<secret>&agent=<agent>&action=match_count

Response:

List of tickets.

2.15 Fetch

Request:

action=fetch

Example:

http://<host>/api.aspx?secret=<secret>&agent=<agent>&action=fetch

Response:

List of tickets.

2.16 Mark_Fetched

Request:

action=mark_fetched

Example:

http://<host>/api.aspx?secret=<secret>&agent=<agent>&action=mark_fetched&fetch_ids=123,124,125

Response:

```
// Errors:
//          1, <error>
// Result:
```

2.17 Check_VenTran (For One Wallet use)

Request:

action=check_ventran

Example:

http://<host>/api.aspx?secret=<secret>&agent=<agent>&action=check_ventran&ventran_id=ABCD123456

Response:

```
// Errors:
//          1, <error>
// Result:
```

Fetch Ticket Info

Ticket

```
xml += "<fid>" + FetchId + "</fid>";                                    // Fetch Id (empty if not action=fetch)
xml += "<id>" + RefNo + "</id>";                                        // Ticket Id
xml += "<t>" + Util.DetailDate(LastModifiedDate) + "</t>";              // Last modified date time up to millisecond
xml += "<u>" + Util.StripPrefix(Action.AGENT, UserName) + "</u>";       // Username
xml += "<b>" + Amt + "</b>";                                            // Bet Amount
xml += "<w>" + WinAmt + "</w>";                                         // Win Amount
xml += "<a>" + com_amt() + "</a>";                                      // Commission Amount
xml += "<c>" + SubCommission + "</c>";                                  // Commission % (value = 0.4 means 0.4%)
xml += "<ip>" + BetIP + "</ip>";                                        // IP address
xml += "<league>" + ModuleId + "</league>";                             // League Id
xml += "<home>" + HomeId + "</home>";                                   // Home Id
xml += "<away>" + AwayId + "</away>";                                   // Away Id
xml += "<status>" + DangerStatus + "</status>";                         // N:Auto-Accept, A:Accepted, R:Reject, C:Cancel
                                                                        // RG:RejectGoal, RP:RejectPenalty, RR:RejectRedCard
xml += "<game>" + game() + "</game>";                                   // HDP,1X2,OU,OE,CSR,TG,FLG,HFT,DC,PAR,ORT, PAS
xml += "<odds>" + odds() + "</odds>";                                   // Odds
xml += "<side>" + side() + "</side>";                                   // 1:Home/Over/Odd, 2:Away/Under/Even, X:Draw
xml += "<info>" + info() + "</info>";                                   // Handicap, over/under, correct score, total goal ...etc
xml += "<half>" + half() + "</half>";                                   // 0:Full Time, 1:First Half
xml += "<trandate>" + Util.StdDate(TransDate) + "</trandate>";          // Transaction date
xml += "<workdate>" + Util.StdDate(WorkingDate) + "</workdate>";        // Working date
xml += "<matchdate>" + MatchDate + "</matchdate>";                      // Match date
xml += "<runscore>" + run_score() + "</runscore>";                      // Running Score (empty if not running match)
xml += "<score>" + score() + "</score>";                                // Score (empty if not match over)
xml += "<htscore>" + ht_score() + "</htscore>";                         // Half time score (empty if not match over & HFT)
xml += "<flg>" + FLGRes + "</flg>";                                     // First/last goal result (empty if not match over & not FLG)
xml += "<res>" + Status + "</res>";                             // P:NotMatchOver WA:WinAll LA:LostAll WH:WinHalf LH:LostHalf D:Draw
xml += "<sportstype>" + SportsTypeId + "</sportstype>";                 //SportsType Id
xml += "<oddstype>" + OddsType + "</oddstype>";                         // MY, HK, ID, EU
xml += "<ventransid>" + VenTransId + "</ventransid>";                   // VenTransId
xml += "<matchdatetime>" + Util.StdDate(TodayDate) + "</ matchdatetime >";   //MatchDateTime
xml += "<curcode>" + CurCode + "</curcode>"                     //Currency Code
```

Parlay (PAR)

```
xml += "<parent>" + RefNo + "</parent>";                                    // Parent ticket Id
xml += "<id>" + SocTransParID + "</id>";                                    // Parlay Id
xml += "<u>" + Util.StripPrefix(Action.AGENT, UserName) + "</u>";           // Username
xml += "<league>" + ModuleId + "</league>";                                 // League Id
xml += "<home>" + HomeId + "</home>";                                       // Home Id
xml += "<away>" + AwayId + "</away>";                                       // Away Id
xml += "<status>" + DangerStatus + "</status>";                             // N:Auto-Accept, A:Accepted, R:Reject, C:Cancel
xml += "<game>" + game() + "</game>";                                       // HDP,1X2,OU,OE,CSR,TG,FLG,HFT,PAR,ORT
xml += "<odds>" + odds() + "</odds>";                                       // Odds
xml += "<side>" + side() + "</side>";                                       // 1:Home/Over/Odd, 2:Away/Under/Even, X:Draw
xml += "<info>" + info() + "</info>";                                       // Handicap, over/under, correct score, total goal ...etc
xml += "<half>" + half() + "</half>";                                       // 0:Full Time, 1:First Half
xml += "<matchdate>" + MatchDate + "</matchdate>";                          // Match date
xml += "<runscore>" + run_score() + "</runscore>";                          // Running Score (empty if not running match)
xml += "<score>" + score() + "</score>";                                    // Score (empty if not match over)
xml += "<res>" + Status + "</res>";                                 // P:NotMatchOver WA:WinAll LA:LostAll WH:WinHalf LH:LostHalf D:Draw
xml += "<matchdatetime>" + Util.StdDate(TodayDate) + "</ matchdatetime >";   //MatchDateTime
```

STEP (PAS)

```
xml += "<parent>" + RefNo + "</parent>";                                    // Parent ticket Id
xml += "<id>" + SocTransPasID + "</id>";                                    // Step Id
xml += "<u>" + Util.StripPrefix(Action.AGENT, UserName) + "</u>";           // Username
xml += "<league>" + ModuleId + "</league>";                                 // League Id
xml += "<home>" + HomeId + "</home>";                                       // Home Id
xml += "<away>" + AwayId + "</away>";                                       // Away Id
xml += "<status>" + DangerStatus + "</status>";                             // N:Auto-Accept, A:Accepted, R:Reject, C:Cancel
xml += "<game>" + game() + "</game>";                                       // HDP,1X2,OU,OE,CSR,TG,FLG,HFT,PAR,ORT
xml += "<odds>" + odds() + "</odds>";                                       // Odds
xml += "<side>" + side() + "</side>";                                       // 1:Home/Over/Odd, 2:Away/Under/Even, X:Draw
xml += "<info>" + info() + "</info>";                                       // Handicap, over/under, correct score, total goal ...etc
xml += "<half>" + half() + "</half>";                                       // 0:Full Time, 1:First Half
xml += "<matchdate>" + MatchDate + "</matchdate>";                          // Match date
xml += "<runscore>" + run_score() + "</runscore>";                          // Running Score (empty if not running match)
xml += "<score>" + score() + "</score>";                                    // Score (empty if not match over)
xml += "<res>" + Status + "</res>";                                 // P:NotMatchOver WA:WinAll LA:LostAll WH:WinHalf LH:LostHalf D:Draw
xml += "<matchdatetime>" + Util.StdDate(TodayDate) + "</ matchdatetime >";   //MatchDateTime
```

1. HDP – Handicap

<side> 1=Home, 2=Away

<info> Handicap, minus means home give. Ex: +2.5, -0.75

2. 1X2 – 1x2

<side> 1=Home, 2=Away, X=Draw

3. OU – Over / Under

<side> 1=Over, 2=Under

<info> Total goal

4. OE – Odds / Even

<side> 1=Odd, 2=Even

5. CSR – Correct Score

<info> Correct score. Ex: 2-1, 0-0

6. TG – Total Goal

<info> Total goal

7. FLG– First Goal / Last Goad

<info> NG=No goal, HLG=Home last goal, ALG=Away last goal, HFG=Home first goal, AFG=Away first goal

<flg> First goal / last goal result

8. HFT – Half Time / Full Time

<info> DD, DH, DA, HD, HH, HA, AD, AH, AA (D=Draw, H=Home, A=Away)

<htscore> Half time score

9. PAR – Parlay

Details in sub-tickets.

10. ORT – Outright

<league> Bet description, ex: NBA Champion

<home> Bet choice, ex: LA Lakers

11.Double Chance

<info> X2, 1X, 12 (X=Draw, 1=Home, 2=Away)

12. PAS – STEP

Details in sub-tickets.