

Saba Integration API

Content

History.....	3
Introduction	10
Terms & Definition	11
Integration Pre-Requirement	12
1. <i>Provide Information</i>	12
2. <i>API Security</i>	12
3. <i>GZip compression and decompression</i>	12
4. <i>Single Wallet Integration Testing Flow</i>	14
Single Wallet API Functions	18
<i>API provided by merchant</i>	18
0. Integration Process	18
1. Get Balance	19
2. Place Bet	22
3. Confirm Bet	35
4. Cancel Bet	39
5. Settle	43
6. Resettle	47
7. Unsettle	51
8. Place Bet Parlay	54
9. Confirm Bet Parlay	66
10. Place Bet 3rd	73
11. Confirm Bet 3rd	80
12. Place Bet Ent	84
13. Settle Ent	88
14. Cancel Bet Ent	92
15. Get Ticket Info	95
16. Health Check	98
17. Adjust Balance	101

API provided by game provider	105
0. Integration Process	105
1. Check Ticket Status	106
2. Retry Operation	111
3. Get Reach Limit Trans	113
4. Resubmit Process	120
5. Check User Outstanding	122
Appendix	124
1. Bet Flow	124
2. Settlement Flow	126
3. Ent Series Flow	127
4. State Diagram	128
5. Supported Products & API list	129
6. Performance and Timeout	130
7. Retry Mechanism	132
8. Status Code	133
9. Combo Type	135
10. CancelBet Error Message	139

History

Version	Description	Modifier	Date
2.0.0.1	Move “operationId” to message level in Settle/Resettle/Unsettle	Jason Ni	2020-09-29
2.0.0.2	Add 3 fields in ConfirmBet: odds, actualAmount, isOddsChanged	Jason Ni	2020-09-29
2.0.0.3	Add oddsType to ConfirmBet	Jason Ni	2020-09-30
2.0.0.4	CancelBet parameter changed from “licenseetxId” to “refId”	Shannon	2020-10-14
2.0.0.5	1. Add one new status code 507, modify status code 311 definition 2. Make character cases consistently in examples	Shannon	2020-10-16
2.0.0.6	1. Add ComfirmBetParlay 2. Add tsId to PlaceBet, PlaceBetParlay	Jason Ni	2020-10-23
2.0.0.7	1. Correct the example of PlaceBetParlay 2. Update State Diagram	Jason Ni	2020-12-09
2.0.0.8	1. Add description for retry mechanism (p.5) 2. Add msg field in GetBalance response	Shannon	2020-12-23
2.0.0.9	PlaceBetParlay: operationId should be in message level, not txns level. Modified document only.	Shannon	2021-01-21
2.0.1.0	Add CashOut	Jessie	2021-02-26
2.0.1.1	Add new parameter: beforeDiscountAmount, percentage on Place Bet 3 rd	Jason Ni	2021-03-31
2.0.1.2	Add PlaceBet3rd, ConfirmBet3rd	Jason Ni	2021-04-01
2.0.1.3	Add new parameters on PlaceBet and PlaceBetParlay baStatus, point, point2, betTeam, homeScore, awayScore, excluding	Jessie	2021-4-21
2.0.1.4	Add new parameters on PlaceBet and PlaceBetParlay leagueName_en, sportTypeName_en,	Jessie	2021-5-19

	betTypeName_en, homeName_en, awayName_en		
2.0.1.5	Add new parameter on CashOut: cashOutTxId Add new parameters and adjust on PlaceBet3rd	Jessie	2021-5-26
2.0.1.6	Modify description for retry mechanism.	Jessie	2021-5-27
2.0.1.7	1. Change “refId” data type from long to string. 2. Adjust “refId” and “operationId” value in examples. 3. Change “licenseetxId” data type from long to string 4. Adjust the description of “beforeDiscountAmount” in document.	Jessie	2021-6-8
2.0.1.8	1. Update State Diagram 2. Update the description of homeScore / awayScore on Place Bet.	Jessie	2021-6-23
2.0.1.9	1. Adjust the description of Performance and Timeout and Retry Mechanism. 2. Add response parameter “balance” on PlaceBet3rd. 3. Add new parameter “betFrom” on PlaceBet, PlaceBetParlay, PlaceBet3rd 4. Add parameter “refId” on ConfirmBet, ConfirmBetParlay, Settle, Resettle, Unsettle, CashOut, ConfirmBet3rd 5. Add parameter “oddsType” on PlaceBetParlay, ConfirmBetParlay, PlaceBet3rd 6. Add new parameters “productName_en” and “gameName_en” on Placebet3rd 7. Add new parameters “creditAmount” and “debitAmount”, on PlaceBet, ConfirmBet, PlaceBetParlay, ConfrimBetParlay, CancelBet, Settle, Resettle, Unsettle, CashOut, PlaceBet3rd, ConfirmBet3rd	Jessie	2021-7-7

2.0.2.0	1. Add integration flow chart: Bet Flow and Settlement Flow. 2. Update the description of retry mechanism. Document modified only.	Jessie	2021-7-23
2.0.2.1	1. Add parameters “creditAmount” and “debitAmount” in txns level on PlaceBetParlay 2. Adjust Cash Out API spec. 2. Add cash out into integration flow chart and state diagram. 3. Remove beforeDiscountAmount and percentage from PlaceBetParlay. 4. Add new parameter oddsInfo on Placebet. 5. Add new parameter betChoice_en on PlaceBet, PlaceBetParlay, PlaceBet3rd. 7. Remove betChoice from ConfirmBetParlay, PlaceBet3rd. 6. Add “Mandatory” column to mark that parameter is required or not. 7. Add new API CashOutResettle	Jessie	2021-8-3
2.0.2.2	1. Change the title “API Pre-Requirement” to “Integration Pre-Requirement”. 2. Add interactive chart and adjust the description on Terms & Definition. 3. Adjust the order and add Integration Process on Single Wallet API Functions. 4. Move Performance & Timeout and Retry Mechanism which were formatted to Appendix (p.76, 77) 5. Move Integration Flow Chart and State Diagram to Appendix. (p.74, 75) 6. Separate State Diagram into two language: English and Chinese. 7. Add hyperlink to check Status Code and Combo Type.	Harry	2021-09-03

	8. Update parameter “parlayType” on “Place Bet Parlay” : Parlay_Combo to Parlay_Mix 9. Add new parameter “matchDateTime” on PlaceBet, PlaceBetParlay 10. Add new parameters “transactionTime” and “winLostDate”, on ConfirmBet, ConfrimBetParlay, ConfirmBet3rd 11. Add new info “parlay detail” on Settle, Resettle, Unsettle 12. Add new API UpdateBet		
2.0.2.3	1. Adjust winLostDate to winlostDate on confirmbet, confirmbetparlay, confirmbet3rd 2. Add new parameter “odds” for Parlay_Mix on placebetparlay and comfirbetparlay 3. Adjust Settle API timeout from 10 seconds to 30 seconds.	Jessie	2021-10-19
2.0.2.4	1. Add new API Health Check	Harry	2021-11-09
2.0.2.5	1. Add new API Check Ticket Status. 2. Add new API Retry Operation. 3. Update the description in parameter “action” 4. Add the description “Gzip compression and decompression” in Integration Pre-Requirement	Harry	2021-12-28
2.0.2.6	1. Remove API Cash Out, Cash Out Resettle, Update Bet. 2. Supports Cash Out feature. Add new parameter “extraStatus” on Settle, Resettle, Unsettle.	Jessie	2022-2-7
2.0.2.7	1. Add new parameter “betRemark” on placebet and PlaceBetParlay	Jessie	2022-3-11
2.0.2.8	1. Remove “parlayDetail” from Settle, Resettle, Unsettle	Jessie	2022-3-18
2.0.2.9	1. Adjust “extraStatus” for void cashout action case on Settle, Resettle, Unsettle.	Jessie	2022-4-8

2.0.3.0	1.Add new APIs : PlaceBetEnt, SettleEnt, Check Ticket Status Ent 2.Add new API Get Reach Limit Trans	Harry	2022-4-8
2.0.3.1	1. Adjust APIs: PlaceBetEnt, SettleEnt. 2. Add new APIs: CancelEnt, GetTicketInfo 3. Add new API: ResubmitProcess 4. Remove API: CheckTicketStatusEnt 5. Add Vgaming Flow, Supported Products & API list	Jessie	2022-6-20
2.0.3.2	1. Add new APIs: AdjustBalance.	Jessie	2022-7-1
2.0.3.3	1.Adjust the description of "oddsType", "betRemark" on PlaceBet and ConfirmBet API 2.Adjust mandatory to N from Y for "betChoice", "betChoice_en" on PlaceBet API	Jessie	2022-11-30
2.0.3.4	1. Support Arcadia Gaming	Jessie	2022-12-23
2.0.3.5	1. Add new status code:508	Jessie	2023-1-12
2.0.3.6	1. Support AdvantPlay and AdvantPlay Mini 2. Adjust description of error code 508 3. Adjust description of Retry Operation API	Jessie	2023-3-8
2.0.3.7	1. Add new parameter "vendorTransId" on Placebet and PlaceBetParlay	Jessie	2023-4-27
2.0.3.8	1. Adjust description of error code 508 23. Adjust description of Retry Operation API	Jessie	2023-6-1
2.0.3.9	1. Add " 4.Single Wallet Integration Testing Flow" under Integration Pre-Requirement	Jessie	2023-6-20
2.0.4.0	1. Add new parameter "mmrPercentage" on Placebet. 2. Add new parameters "mmrPercentage" and "isMmrPercentageChange" on Confirmbet.	Jessie	2023-7-12
2.0.4.1	1. Support MaxGame 2. Remove AdvantPlay and AdvantPlay Mini	Jessie	2023-9-26

2.0.4.2	1. Add new parameter "errorMessage" on CancelBet. 2. Add new Appendix: CancelBet Error Message	Jessie	2023-11-10
2.0.4.3	1. Add new parameter "extraInfo", "isOnlyWinlostDateChanged" on resettle API. 2. Update the description of resettle API. Document modified only.	Jessie	2023-11-27
2.0.4.4	1. Add new parameter "refNo" on AdjustBalance	Jessie	2023-12-29
2.0.4.5	1. Remove Arcadia Gaming, MaxGame products.	Jessie	2024-1-11
2.0.4.6	1. Add new parameter "voucher" on PlaceBet. 2. Add new error message on Appendix- CancelBet Error Message.	Jessie	2024-3-7
2.0.4.7	1. Provide more detailed information about the operationId. Document modified only.	Jessie	2024-6-24
2.0.4.8	1. Add bettype 17006 in AdjustBalance description.	Jessie	2024-7-17
2.0.4.9	1. Add new API: CheckUserOutstanding.	Jessie	2024-8-5
2.0.5.0	1. Add new parameter "settlementTime" on Settle and Resettle API.	Jessie	2024-8-23
2.0.5.1	1. Add new parameter "odds" on Settle and Resettle API.	Jessie	2024-9-6
2.0.5.2	1. Adjust Appendix- Supported Products & API list 2. Add parameters in PlaceBet.voucher list	Jessie	2024-10-23
2.0.5.3	1. Add voucher type: Odds Boost voucher (Giveaway) on PlaceBet API.	Jessie	2024-11-13
2.0.5.4	1. Add new parameter "gameGroup" on PlaceBet.	Lydia	2024-12-19
2.0.5.5	1. Modify the licenseeTxId example, changing the type to string.	Lydia	2025-01-10
2.0.5.6	1. Cancel Bet Return Parameter Status / Trigger Frequency and Count Limit - Additional Explanation	Lydia	2025-03-07

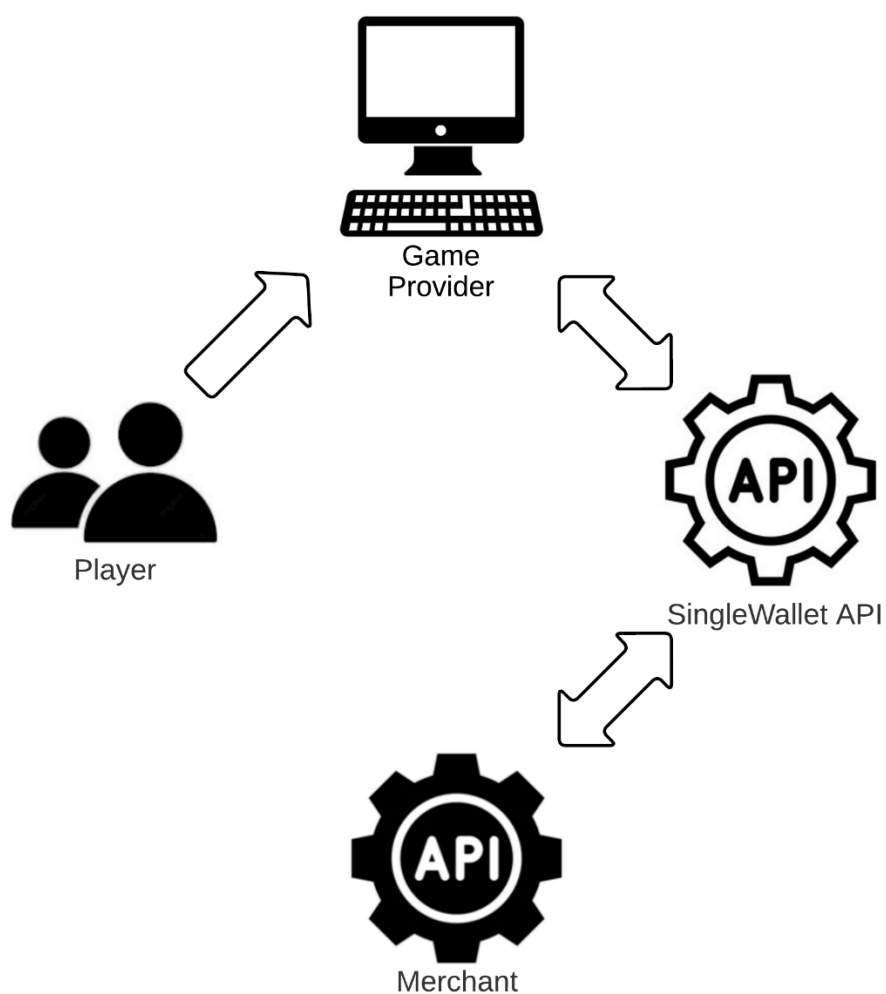
2.0.5.7	1. Add bettype 17011 & 17012 in AdjustBalance description.	Harry	2025-03-19
2.0.5.8	1. Place Bet: Remove parameter mmrPercentage 2. Confirm Bet: Remove parameters mmrPercentage and isMmrPercentageChange	Lydia	2025-04-11

Introduction

This document is to provide the technical specs for our Merchant to perform integration and test with Saba games. Including functions, parameters, return results and error codes.

Terms & Definition

Name	Description
Merchant	Saba Partner who operates its own gaming and who would like to integrate with Saba Sports.
Game Provider	Saba, to provide game API for Merchant.
Player	Merchant's end client. A player is an end-user of web-application, who can access web-application games via Merchant's website.
Single Wallet API	This is an API which provides an interface between Merchant and Game Provider and has a full control on transactions.



Integration Pre-Requirement

1. Provide Information

- Server Info & API URL
- Credential Details: Key (vendor_id)

2. API Security

- Need to setup white IP.
- Need HTTPS for Production Environment.

3. GZip compression and decompression

- Every request's body will be compressed by GZip in all API provided by merchant.
- Need to decompress by GZip.
- Here is the sample code for .NET.
 - Need to register "DecompressionHandler" at method "Register" in WebApiConfig

```

1. public static void Register(HttpConfiguration config)
2. {
3.     config.MapHttpAttributeRoutes();
4.     config.Routes.MapHttpRoute(
5.         name: "DefaultApi",
6.         routeTemplate: "api/{controller}/{action}/{id}",
7.         defaults: new { id = RouteParameter.Optional }
8.     );
9.     config.MessageHandlers.Add(new DecompressionHandler());
10. }
```

- Add the following method after method "Register".

```

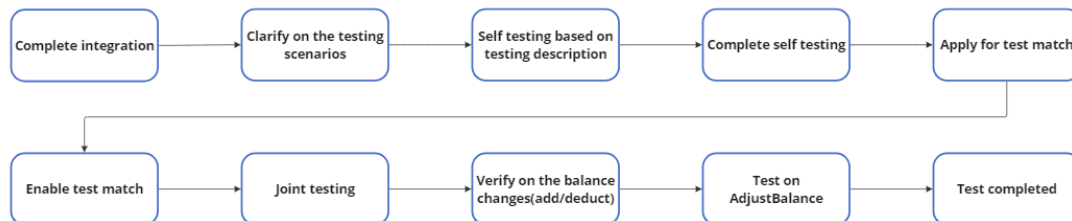
1. /// <summary>
2. /// Decompression GZIP, Deflate content
3. /// </summary>
4. /// <seealso cref="System.Net.Http.DelegatingHandler" />
5. public class DecompressionHandler : DelegatingHandler
6. {
7.     protected override async Task<HttpResponseMessage> SendAsync(HttpRequestMessage request,
8.         CancellationToken cancellationToken)
```

```
9.      {
10.         if (request.Method == HttpMethod.Post)
11.         {
12.             bool isGzip = request.Content.Headers.ContentEncoding.Contains("gzip");
13.             if (isGzip)
14.             {
15.                 Stream decompressedStream = new MemoryStream();
16.                 using (var gzipStream = new GZipStream(await request.Content.ReadAsStreamAsync(),
17.                     CompressionMode.Decompress))
18.                 {
19.                     await gzipStream.CopyToAsync(decompressedStream);
20.                 }
21.                 decompressedStream.Seek(0, SeekOrigin.Begin);
22.
23.                 var originContent = request.Content;
24.                 request.Content = new StreamContent(decompressedStream);
25.
26.                 foreach (var header in originContent.Headers)
27.                 {
28.                     request.Content.Headers.Add(header.Key, header.Value);
29.                 }
30.             }
31.         }
32.         return await base.SendAsync(request, cancellationToken);
33.     }
34. }
```

4. Single Wallet Integration Testing Flow

First and foremost, thanks for supporting SABA!

Before we provide production environment, you will need to run a thorough test on Single Wallet APIs. Once it is completed and verified, we will apply production credentials for you.



Please make sure below implementation is complete before you start testing:

- Endpoints that need to be implemented (Different products will need to implement different endpoints)

Merchant API (Mandatory)

[CreateMember], [GetSabaURL], [CheckTicketStatus], [RetryOperation], [GetReachLimitTrans]

Sportsbook and Cashout (Mandatory)

[GetBalance], [PlaceBet], [ConfirmBet], [PlaceBetParlay] , [ConfirmBetParlay] , [Settle] , [Resettle] , [Unsettle] , [CancelBet] , [AdjustBalance]

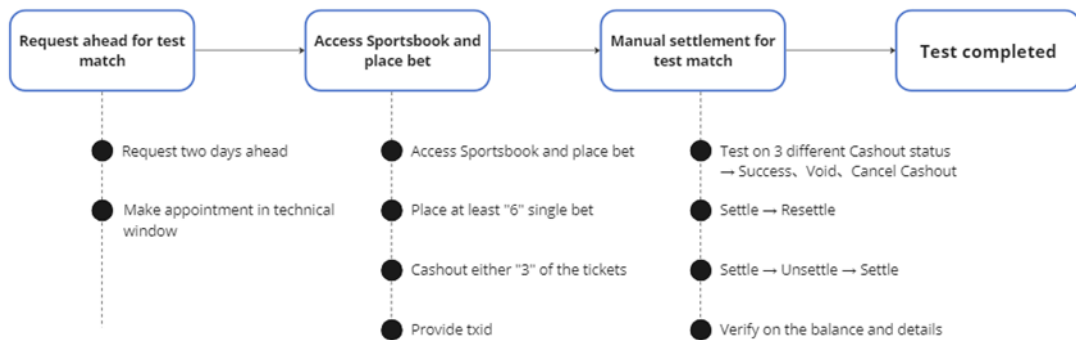
- If you have following products : Keno / Lottery / Table Games
[PlaceBet3rd] \ [ConfirmBet3rd]

If you have following products : Vgaming

[PlaceBetEnt] \ [SettleEnt] \ [CancelBetEnt] \ [GetTicketInfo] \ [ResubmitProcess]

- All of the single wallet requests' body are compressed with [GZIP](#), please make sure to decompress the requests in order to process on it. (Please refer to "[GZip compression and decompression](#)" in our API document)
- Please provide callback URL for us to setup once your callback service is ready.
Kindly note: Should you need to whitelist our server outgoing IPs, please contact Saba Technical Window during working hours (Monday to Friday 10:00~18:00).
- You can run a preliminary test on your own once setup is done (Kindly refer to below description, and run the test on your own).

- **Testing Flow:**



- **Test description (Run by merchant)**

How to test for GetBalance?

1. Please trigger api/GetSabaUrl to generate login url. We will send GetBalance request upon login to Sportsbook.
2. For desktop platform, click on refresh button next to current balance will receive Getbalance request.

How to test for PlaceBet?

We will send PlaceBet request when place bet action is carried out.

How to test for ConfirmBet?

We will send Confirmbet request once we receive "Success" response for Placebet request

How to test for CancelBet?

We will send CancelBet request if PlaceBet request timed out (Time out = 3 seconds)

How to test for Settle?

We will send Settle request once match is completed and settled

PS. For quick settlement, please place bet on SABA Virtual matches

How to test for PlaceBetParlay & ConfirmBetParlay?

Click on "Parlay", then select few matches and bet on different Parlay combinations. We will send PlaceBetParlay request, and once we received "Success" response, will proceed to send

ConfirmBetParlay.

How to test for CheckTicketStatus & RetryOperation?

Step 1: Generate a failed request, such as ConfirmBet or Settle, by responding us with an error code that will not invoke retry mechanism (Please refer to "[Retry Mechanism](#)" in our API document).

Step 2: Trigger CheckTicketStatus by using either Refid or txid of the ticket, and retrieve the OperationID of the failed request.

Step 3: Trigger RetryOperation with the OperationID of the failed request and check whether you receive the request.

- **Testing description (Between merchant and Saba)**

If aforementioned test is completed, please inform Saba technician via technical window to verify on the logs. Once it is verified, we will arrange a test match to test on below scenarios.

Note: Please inform Saba technician two days ahead and we will inform on the date and time. Kindly proceed to bet within 30 minutes once test match is provided. Otherwise, test match will be closed and will need to re-arrange another time for test match again.

Endpoints that will be triggered during the testing flow:

[GetBalance] \ [PlaceBet] \ [ConfirmBet] \ [Settle] \ [Resettle] \ [Unsettle]

Flow:

Step1: Saba technicians will provide test match on the appointed date.

Step2: Merchant will need to access Sportsbook and place bets. Please place at least 6 single bets, and Cashout either 3 of the single bets and provide us with txid for manual settlement.

Step3: We will test on Cashout bets (Status: Success, void, cancel cashout), merchant will need to verify on the balance for these 3 kinds of status before continuing to test on single bets.

Step4: We will settle the match and test on below scenarios. Merchant will need to verify on the balance changes.

1. Settle → Resettle
2. Settle → Unsettle → Settle

Step5: Test complete

How to test for AdjustBalance?

After all the test is completed, please provide us one test account to test on AdjustBalance. We will manually send an AdjustBalance request, and you will have to make sure the balance is added correctly.

Single Wallet API Functions

API provided by merchant

0. Integration Process

The following steps are recommended order to apply Single Wallet API.

First Step



To complete single bet and cash out

- | | |
|----------------|-------------|
| 1. Get Balance | 5. Settle |
| 2. Place Bet | 6. Resettle |
| 3. Confirm Bet | 7. Unsettle |
| 4. Cancel Bet | |

Second Step



To complete parlay bet and 3rd-party game

- | | |
|-----------------------|---------------------|
| 8. Place Bet Parlay | 12. Place Bet Ent |
| 9. Confirm Bet Parlay | 13. Settle Ent |
| 10. Place Bet 3rd | 14. Cancel Bet Ent |
| 11. Confirm Bet 3rd | 15. Get Ticket Info |

Third Step



To check connection stability

16. Health Check

Fourth Step



To complete promotional campaign

17. Adjust Balance

1. Get Balance

- Description

- ✧ Game provider is able to call this function that will be provided by merchant.
- ✧ Game provider need to check player latest balance regularly and display on the game site.

- Method URL

Provide by Merchant.

For Example: https://<Operator_API_SERVICE_URL>/getbalance

- Request Parameter

Field	Type	Mandatory	Description
key	string	Y	Value of key in credential document
message	json	Y	Json Format: reference to table below.

JSON Information:

Field	Mandatory	Description
action	Y	(string)GetBalance
userId	Y	(string)user id

For example:

```
{
  "key": "xxx",
  "message": {
    "action": "GetBalance",
    "userId": "player"
  }
}
```

- Response Parameter

Field	Mandatory	Type	Description
status	Y	string	status code (Appendix: Status Code)
userId	Y	String	user id
balance	Y	decimal	If status is success that return balance for update.
balanceTs	Y	string	If status is success that return balanceTs for update(ISO8601 format) GMT-4 YYYY-MM-DDThh:mm:ss.sss Please ensure balance is up-to-date whenever the balance is changed.
msg	N	string	If status is failed, then return description for this failure.

Successful Result

```
{
  "status": "0",
  "userId": "player",
  "balance": 100.00,
  "balanceTs": "2018-10-01T09:43:22.959-04:00"
  "msg": null
}
```

Failed Result

```
{
  "status": "203",
  "msg": "Account is not exist"
}
```

```
}
```

2. Place Bet

- Description

Game provider is able to call this function that will be provided by merchant, and send out the bet details to merchant.

- ✧ Whenever the failed to place bet on Saba system, game provider will call the Cancel Bet API to cancel the reserve bet.
- ✧ Please trigger CheckTicketStatus to verify the ticket status if you did not receive ConfirmBet or Cancelbet within 10 minutes after PlaceBet
- ✧ If doesn't receive response, game provider will also call the Cancel Bet API to cancel the reserve bet.
- ✧ **The operationId is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.**

- Method URL

Provide by Merchant.

For Example: https://<Operator_API_SERVICE_URL>/placebet

- Request Parameter

Field	Type	Mandatory	Description
key	string	Y	Value of key in credential document.
message	json	Y	Json Format: reference to table below.

JSON Information:

Field	Mandatory	Description
action	Y	(string) PlaceBet
operationId	Y	(string) It is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.
userId	Y	(string) User id
currency	Y	(int) Saba Currency Id e.g. 1, 2, 20 Please refer to Appendix: Currency Table

matchId	Y	(int) e.g. 35627959
homeId	Y	(int) e.g. 23, 24
awayId	Y	(int) e.g. 23, 24
homeName	Y	(string) This value depends on player's language. e.g. Chile (V)
awayName	Y	(string) This value depends on player's language. e.g. France (V)
kickOffTime	Y	(string) Kick off time (yyyy-MM-dd HH:mm:ss.SSS) GMT-4
betTime	Y	(string) Place bet time (yyyy-MM-dd HH:mm:ss.SSS) GMT-4
betAmount	Y	(decimal) Input bet amount
actualAmount	Y	(decimal) Actual bet amount
sportType	Y	(int) e.g. 1, 2, 3 Please refer to Appendix: Sports Table
sportTypeName	Y	(string) This value depends on player's language. e.g. Soccer
betType	Y	(int) e.g. 1, 3 Please refer to Appendix: BetType Table
betTypeName	Y	(string) This value depends on player's language. e.g. Handicap
oddsType	Y	(short) This Value will be 0 on bettype=8700(Player Tips), 468,469. e.g. 0,1, 2, 3, 4, 5,6 Please refer to Appendix: OddsType Table
oddsId	Y	(int) e.g. 246903111
odds	Y	(decimal) e.g. -0.95, 0.75
betChoice	N	(string) e.g. Over, 4-3
betChoice_en	N	(string) BetChoice in English. e.g. Over, 4-3
updateTime	Y	(string) Update time when bet has changed. (yyyy-MM-dd HH:mm:ss.SSS) GMT-4
leagueId	Y	(int) e.g. 152765
leagueName	Y	(string) This value depends on player's language. e.g. SABA INTERNATIONAL FRIENDLY Virtual PES 20 - 20 Mins Play
leagueName_en	Y	(string) League name in English. e.g. SABA INTERNATIONAL FRIENDLY Virtual PES 20 - 20 Mins Play
sportTypeName_en	Y	(String) Sport type name in English. e.g. Soccer

betTypeName_en	Y	(String) Bet type name in English. e.g. Handicap
homeName_en	Y	(String) Home name in English. e.g. Chile (V)
awayName_en	Y	(String) Away name in English. e.g. France (V)
IP	Y	(string) e.g. 61.221.35.49 (IPv4)
isLive	Y	(boolean) e.g. true, false
refId	Y	(string) Unique id.
tsId	N	(string) optional, user login session id sent by merchant
point	N	(string) Line. For number game (sporttype=161),It's current number before betting.
point2	N	(string) Line2. Only bettype 646 with value, point is HDP and point2 is OU.
betTeam	N	(string)The team/side that customer placed bet on
homeScore	N	(int) The home team's score when placing bet. Total ball count of the ball number greater than 37.5 for number game (sporttype=161). e.g.1
awayScore	N	(int) The away team's score when placing bet Total ball count of the ball number less than 37.5 for number game (sporttype=161). e.g.1
baStatus	Y	(boolean)Flag of Bet Aggregator true/false.
excluding	N	(string)Only returned when bet_team=aos. The returned value means that the correct score player placed on is not the listed options
betFrom	Y	(string) The platform in which the bet came from. Please refer to Appendix: BetFrom Table
creditAmount	Y	(decimal) The amount should be credited on player.
debitAmount	Y	(decimal) The amount should be deducted from player.
oddsInfo	N	(string) Only bettype 468,469 with value.
matchDateTime	Y	(string) Match date time (yyyy-MM-dd HH:mm:ss.SSS) GMT-4 Note: matchDatetime of Outright Betting will be KickOffTime.

betRemark	N	(string) -bettype 9662~9667(Cricket), 9676~9705(Cricket) with value. Bet type: 9662~9667 Will display two players name Bet type: 9676~9705 Will display one player name -bettype 8700(Player Tips) with value to show streamer code and language. e.g. ss=20-(CN) , ss=21-(VN), ss=22-(TH) ss is Streamer source. Streamer code is 22. Language is TH.
vendorTransId	N	(string) This parameter will be returned only if the bet is placed through Direct API and Odds Feed API.
voucher	N	(List)
type	N	(int) Voucher type. 1: Risk Free Bet voucher. 2: Odds Boost voucher 4: Odds Boost voucher (Giveaway): Special events only.
quota	N	(decimal) Voucher quota. Will show when type=1: Risk Free Bet voucher.
beforeOddsBoost	N	(decimal) The odds which is before using odds boost voucher. Will show when type 2: Odds Boost voucher. 4: Odds Boost voucher (Giveaway)
oddsBoostPoint	N	(decimal) The odds boost point of voucher. Will show when type 2: Odds Boost voucher. 4: Odds Boost voucher (Giveaway)
gameGroup	Y	(String) Differentiates the type of virtual sports event Return values: Saba virtual sports, Saba pingol Others: null

For example:

```
{
  {
```

```

"key": "XXX",
"message": {
  "action": "PlaceBet",
  "operationId": "4200500_1_924",
  "userId": " player ",
  "currency": 20,
  "matchId": 43541666,
  "homeId": 77343,
  "awayId": 68311,
  "homeName": "Blaublitz Akita",
  "awayName": "Renofa Yamaguchi",
  "kickOffTime": "2021-05-19T05:29:59.000-04:00",
  "betTime": "2021-05-18T22:48:07.252-04:00",
  "betAmount": 2,
  "actualAmount": 1.92,
  "sportType": 1,
  "sportTypeName": "Soccer",
  "betType": 3,
  "betTypeName": "Over/Under",
  "oddsType": 1,
  "oddsId": 337051838,
  "odds": -0.96,
  "betChoice": "Over",
  "betChoice_en": "Over",
  "updateTime": "2021-05-18T22:48:07.252-04:00",
  "leagueId": 344,
  "leagueName": "JAPAN J-LEAGUE DIVISION 2",
  "leagueName_en": "JAPAN J-LEAGUE DIVISION 2",
  "sportTypeName_en": "Soccer",
  "betTypeName_en": "Over/Under",
  "homeName_en": "Blaublitz Akita",

```

```

    "awayName_en": "Renofa Yamaguchi",
    "IP": "61.221.35.49",
    "isLive": false,
    "refId": "4200500_337",
    "tsId": "",
    "point": "2.00",
    "point2": "",
    "betTeam": "h",
    "homeScore": 0,
    "awayScore": 0,
    "baStatus": false,
    "excluding": "",
    "betFrom": "x",
    "creditAmount": 0,
    "debitAmount": 1.92,
    "oddsInfo": ""
    "matchDateTime": "2021-05-19T05:30:59.000-04:00",
    "betRemark": "",
    "vendorTransId": "test20230427001",
    "gameGroup": "Saba virtual sports",
    "voucher": [
        {
            "type": 1,
            "quota": 1.92
        }
    ]
}

```

✧ For Saba Reward Voucher

Place bet with using Risk Free Bet voucher.

```

{
  {
    "key": "XXX",
    "message": {
      "action": "PlaceBet",
      "operationId": "4200500_1_924",
      "userId": " player ",
      "currency": 20,
      "matchId": 43541666,
      "homeId": 77343,
      "awayId": 68311,
      "homeName": "Blaublitz Akita",
      "awayName": "Renofa Yamaguchi",
      "kickOffTime": "2021-05-19T05:29:59.000-04:00",
      "betTime": "2021-05-18T22:48:07.252-04:00",
      "betAmount": 2,
      "actualAmount": 1.92,
      "sportType": 1,
      "sportTypeName": "Soccer",
      "betType": 3,
      "betTypeName": "Over/Under",
      "oddsType": 1,
      "oddsId": 337051838,
      "odds": -0.96,
      "betChoice": "Over",
      "betChoice_en": "Over",
      "updateTime": "2021-05-18T22:48:07.252-04:00",
      "leagueId": 344,
      "leagueName": "JAPAN J-LEAGUE DIVISION 2",
      "leagueName_en": "JAPAN J-LEAGUE DIVISION 2",
      "sportTypeName_en": "Soccer",

```

```

    "betTypeName_en": "Over/Under",
    "homeName_en": "Blaublitz Akita",
    "awayName_en": "Renofa Yamaguchi",
    "IP": "61.221.35.49",
    "isLive": false,
    "refId": "4200500_337",
    "tsId": "",
    "point": "2.00",
    "point2": "",
    "betTeam": "h",
    "homeScore": 0,
    "awayScore": 0,
    "baStatus": false,
    "excluding": "",
    "betFrom": "x",
    "creditAmount": 0,
    "debitAmount": 1.92,
    "oddsInfo": ""
    "matchDateTime": "2021-05-19T05:30:59.000-04:00",
    "betRemark": "",
    "vendorTransId": "test20230427001",
    "voucher": [
        {
            "type": 1,
            "quota": 1.92
        }
    ]
}

```

✧ For Saba Reward Voucher

Place bet with using Odds Boost voucher.

```
{
  {
    "key": "XXX",
    "message": {
      "action": "PlaceBet",
      "operationId": "4200500_1_924",
      "userId": " player ",
      "currency": 20,
      "matchId": 43541666,
      "homeId": 77343,
      "awayId": 68311,
      "homeName": "Blaublitz Akita",
      "awayName": "Renofa Yamaguchi",
      "kickOffTime": "2021-05-19T05:29:59.000-04:00",
      "betTime": "2021-05-18T22:48:07.252-04:00",
      "betAmount": 2,
      "actualAmount": 1.92,
      "sportType": 1,
      "sportTypeName": "Soccer",
      "betType": 3,
      "betTypeName": "Over/Under",
      "oddsType": 1,
      "oddsId": 337051838,
      "odds": -0.96,
      "betChoice": "Over",
      "betChoice_en": "Over",
      "updateTime": "2021-05-18T22:48:07.252-04:00",
      "leagueId": 344,
      "leagueName": "JAPAN J-LEAGUE DIVISION 2",
    }
  }
}
```

```

"leagueName_en": "JAPAN J-LEAGUE DIVISION 2",
"sportTypeName_en": "Soccer",
"betTypeName_en": "Over/Under",
"homeName_en": "Blaublitz Akita",
"awayName_en": "Renofa Yamaguchi",
"IP": "61.221.35.49",
"isLive": false,
"refId": "4200500_337",
"tsId": "",
"point": "2.00",
"point2": "",
"betTeam": "h",
"homeScore": 0,
"awayScore": 0,
"baStatus": false,
"excluding": "",
"betFrom": "x",
"creditAmount": 0,
"debitAmount": 1.92,
"oddsInfo": ""
"matchDateTime": "2021-05-19T05:30:59.000-04:00",
"betRemark": "",
"vendorTransId": "test20230427001",
"voucher": [
    {
        "type": 2,
        "beforeOddsBoost": 1.92
        "oddsBoostPoint": 1.92
    }
]
}

```



```
}

```

✧ For Giveaway Voucher

Place bet with using Odds Boost voucher (Giveaway).

```
{
  "key": " XXX ",
  "message": {
    "action": "PlaceBet",
    "operationId": "4282200_1_126066_U",
    "userId": " player ",
    "currency": 20,
    "matchId": 95301684,
    "homeId": 6,
    "awayId": 55,
    "homeName": "Arsenal",
    "awayName": "Nottingham Forest",
    "kickOffTime": "2024-11-23T11:15:00.000-04:00",
    "betTime": "2024-11-13T03:06:47.251-04:00",
    "betAmount": 10.0,
    "actualAmount": 10.0,
    "sportType": 1,
    "sportTypeName": "Soccer",
    "betType": 3,
    "betTypeName": "Over/Under",
    "oddsType": 3,
    "oddsId": 757034955,
    "odds": 2.01,
    "betChoice": "Over",
    "betChoice_en": "Over",
    "updateTime": "2024-11-13T03:06:47.251-04:00",
    "leagueId": 3,
  }
}
```

```

    "leagueName": "*ENGLISH PREMIER LEAGUE",
    "leagueName_en": "*ENGLISH PREMIER LEAGUE",
    "sportTypeName_en": "Soccer",
    "betTypeName_en": "Over/Under",
    "homeName_en": "Arsenal",
    "awayName_en": "Nottingham Forest",
    "IP": "211.75.116.91",
    "isLive": false,
    "refId": "4282200_40569_U",
    "tsId": "",
    "point": "2.75",
    "point2": "",
    "betTeam": "h",
    "homeScore": 0,
    "awayScore": 0,
    "baStatus": false,
    "excluding": "",
    "betFrom": "W002",
    "creditAmount": 0.0,
    "debitAmount": 10.0,
    "oddsInfo": "",
    "matchDateTime": "2024-11-23T11:00:00.000-04:00",
    "betRemark": "",
    "voucher": [
        {
            "type": 4,
            "beforeOddsBoost": 1.99,
            "oddsBoostPoint": 0.02
        }
    ]
}

```

```
}

```

- Response Parameter

Field	Mandatory	Type	Description
status	Y	string	Status code(Appendix: Status Code)
refId	Y	string	Get from the parameters of request.
licenseeTxId	Y	string	Transaction id of merchant system.
msg	N	string	If status is failed, then return description for this failure. (Appendix: Status Code)

Successful Result

```
{
  "status": "0",
  "refId": "4200500_2331",
  "licenseeTxId": "1988754"
}
```

Failed Result

```
{
  "status": "203",
  "msg": "Account is not exist"
}
```

3. Confirm Bet

- Description

Game provider is able to call this function that provided by merchant.

- ✧ When PlaceBet is successful on provider side, game provider will call the ConfirmBet API.
- ✧ Will repeatedly call Confirm Bet until it is successfully implemented or reach the limit in retry progress if it is failed to call in the first attempt. Can refer Appendix: Retry Mechanism for further detail.
- ✧ When the odds is changed to better one, request.isOddsChanged should be true then request.odds and request.actualAmount will be different from PlaceBet.
- ✧ **The operationId is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.**

- Method URL

Provide by Merchant.

For Example: https://<Operator_API_SERVICE_URL>/confirmbet

- Request Parameter

Field	Type	Mandatory	Description
key	string	Y	Value of key in credential document.
message	json	Y	Json Format: reference to table below.

JSON Information:

Field	Mandatory	Description
action	Y	(string) ConfirmBet
operationId	Y	(string) It is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.
userId	Y	(string) user id
updateTime	Y	(string) Update time when bet has changed. (yyyy-MM-dd HH:mm:ss.SSS) GMT-4

transactionTime	Y	(string) The transaction time in which bet was placed in Saba system. (yyyy-MM-dd HH:mm:ss.SSS) GMT-4
txns	Y	(array) Reference to table of <i>Ticket Info</i> .

Ticket Info:

Field	Mandatory	Description
refId	Y	(string) Unique id.
txId	Y	(long) Transaction id of saba system.
licenseeTxId	Y	(string) Transaction id of merchant system.
odds	Y	(decimal) e.g. -0.75, 0.96
oddsType	Y	(short) This Value will be 0 on bettype=8700(Player Tips), 468,469. e.g. 0,1, 2, 3, 4, 5,6 Please refer to Appendix: OddsType Table
actualAmount	Y	(decimal) Actual bet amount
isOddsChanged	Y	(boolean) e.g. true, false
creditAmount	Y	(decimal) The amount should be credited on player.
debitAmount	Y	(decimal) The amount should be deducted from player.
winlostDate	Y	(string) The time in which the win lose is finalized(show date only). (yyyy-MM-dd 00:00:00.000) GMT-4

For example:

```
{
  "key": "xxx",
  "message": {
    "action": "ConfirmBet",
    "operationId": "4200500_1_21",
    "userId": "player",
    "updateTime": "2020-04-01T09:43:22.959-04:00",
    "transactionTime": "2020-04-01T09:44:22.959-04:00",
```

```

    "txns":[
      {
        "refId":"4200500_337",
        "txId":1854,
        "licenseeTxId":"2599635",
        "odds":0.76,
        "oddsc":4,
        "actualAmount":100,
        "isOddsChanged":false
        "creditAmount": 0,
        "debitAmount": 0,
        "winlostDate": "2020-04-02T00:00:00.000-04:00"
      }
    ]
  }
}

```

- Response Parameter

Field	Mandatory	Type	Description
status	Y	string	status code (Appendix: Status Code)
balance	Y	decimal	If status is success, return balance for update. 2 decimals
msg	N	string	If status is failed, then return description for this failure.

Successful Result

```
{  
  "status": "0",  
  "balance": 100.00  
}
```

Failed Result

```
{  
  "status": "203",  
  "msg": "Account is not exist"  
}
```

4. Cancel Bet

- Description

Game provider is able to call this function that will be provided by merchant.

- ✧ When the bet is place unsuccessfully on Saba system, or an unexpected problem leads to Saba does not receive a response from merchant then cause time out, game provider will call cancel bet.
- ✧ Will repeatedly call Cancel Bet until it is successfully implemented or reach the limit in retry progress if it is failed to call in the first attempt. Can refer Appendix: Retry Mechanism for further detail.
- ✧ Support multiple transaction of the same player to be cancelled at the same time.
- ✧ **The operationId is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.**

- Method URL

Provide by Merchant.

For Example: https://<Operator_API_SERVICE_URL>/cancelbet

- Request Parameter

Field	Type	Mandatory	Description
key	string	Y	Value of key in credential document.
message	json	Y	Json format:reference to table below.

JSON Information:

Field	Mandatory	Description
action	Y	(string)CancelBet
operationId	Y	(string) It is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.
userId	Y	(string)User id

updateTime	Y	(string)Update time when bet has changed. (yyyy-MM-dd HH:mm:ss.SSS) GMT-4
errorMessage	Y	(string) The error message of bet failed reason. All possible error messages can be check in the reference list. (Appendix: CancelBet Error Message)
txns	Y	Json Format: reference to table below

****If there are multiple transactions, the transaction information can be sent as an array**

Field	Mandatory	description
refId	Y	(string) Unique id.
creditAmount	Y	(decimal) The amount should be credited on player.
debitAmount	Y	(decimal) The amount should be deducted from player.

For example:

```
{
  "key": "xxx",
  "message": {
    "action": "CancelBet",
    "operationId": "4200500_1_24",
    "userId": "player01",
    "updateTime": "2020-04-01T09:43:22.959-04:00",
    "errorMessage": "BettingFailed"
    "txns": [{
      "refId": "4200500_2331",
      "creditAmount": 1.92,
```

```
        "debitAmount": 0  
    }  
}
```

Response Parameter

Field	Mandatory	Type	Description
status	Y	String	status code (Appendix: Status Code) Status Code: 504 will be retried 10 times, while other status codes will be retried 100 times.
balance	Y	decimal	If status is success, return balance for update. 2 decimals
msg	N	string	If status is failed, then return description for this failure.

Successful Result

```
{
  "status": "0",
  "balance": 100.00
}
```

Failed Result

```
{
  "status": "203",
  "msg": "Account is not exist"
}
```

5. Settle

- Description

Game provider is able to call this function that will be provided by merchant.

- ✧ Game provider will send out the transaction after the game is complete or send out the transaction after the Cash Out is accepted
- ✧ Will repeatedly call Settle until it is successfully implemented or reach the limit in retry progress if it is failed to call in the first attempt. Can refer Appendix: Retry Mechanism for further detail.
- ✧ **The operationId is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.**

- Method URL

Provide by Merchant.

For Example: https://<Operator_API_SERVICE_URL>/settle

- Request Parameter

Field	Type	Mandatory	Description
key	string	Y	value of key in credential document
message	json	Y	Json Format: reference to table below.

JSON Information:

Field	Mandatory	Description
action	Y	(string) Settle
operationId	Y	(string) It is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.
txns	Y	Json Format: reference to table below

****If there are multiple transactions, the transaction information can be sent as an array**

Field	Mandatory	Description
userId	Y	(string) User id.

refId	Y	(string) Unique id.
txId	Y	(long) Transaction id of saba system.
updateTime	Y	Update time when bet has changed. (yyyy-MM-dd HH:mm:ss.SSS) GMT-4
winlostDate	Y	The time in which the win lose is finalized(show date only). (yyyy-MM-dd 00:00:00.000) GMT-4
status	Y	(string) half won/half lose/won/lose/draw/void/refund/reject
payout	Y	(decimal) The winning amount for ticket.
creditAmount	Y	(decimal) The amount should be credited on player.
debitAmount	Y	(decimal) The amount should be deducted from player.
extraStatus	N	(String) Will have value in the following cases. -Ticket has been cash out-> "extraStatus": "cashout". In this case will send via Settle API. -Ticket has been voided after ticket cash out -> "extraStatus": "void ticket". In this case will send via Resettle API. -Match has been refunded after ticket cash out -> "extraStatus": "refund match". In this case will send via Resettle API. -Ticket has been voided cash out action after ticket cash out -> "extraStatus": "void cashout". In this case will send via Unsettle API.
settlementTime	Y	The time in which the ticket is settled. If ticket status=reject ,void, settlementTime will be "null".
odds	N	(decimal) This parameter will show only if bettype=468,469.

For example:

```
{
  "key": "xxx",
  "message": {
    "action": "Settle",
    "operationId": "4200500_1_25",
```

```
"txns": [  
  {  
    "userId": "player01",  
    "refId": "4200500_555",  
    "txId": 339482738748,  
    "updateTime": "2021-09-08T04:49:32.383-04:00",  
    "winlostDate": "2021-09-08T00:00:00.000-04:00",  
    "status": "won",  
    "payout": 3.5600,  
    "creditAmount": 3.5600,  
    "debitAmount": 0.0,  
    "extraStatus": "",  
    "settlementTime": "2021-09-08T04:49:26.3"  
  }  
]  
}
```

- Response Parameter

Field	Mandatory	Type	Description
-------	-----------	------	-------------

status	Y	string	status code (Appendix: Status Code)
msg	N	string	If status is failed, then return description for this failure.

Successful Result

```
{  
  "status": "0"  
}
```

Failed Result

```
{  
  "status": "203",  
  "msg": "Account is not exist"  
}
```

6. Resettle

● Description

Game provider is able to call this function that will be provided by merchant.

- ✧ Game provider will send out the transaction, when one of the winlostDate, status, creditAmount, debitAmount or payout changes due to the match has been re-settled.
- ✧ Will repeatedly call Resettle until it is successfully implemented or reach the limit in retry progress if it is failed to call in the first attempt. Can refer Appendix: [Retry Mechanism](#) for further detail.
- ✧ **The operationId is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.**

● Method URL

Provide by Merchant.

For Example: https://<Operator_API_SERVICE_URL>/resettle

● Request Parameter

Field	Type	Mandatory	Description
key	string	Y	Value of key in credential document.
message	json	Y	Json Format: reference to table below.

JSON Information:

Field	Mandatory	Description
action	Y	(string) Resettle
operationId	Y	(string) It is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.
txns	Y	Json Format: reference to table below

****If there are multiple transactions, the transaction information can be sent as an array**

Field	Mandatory	Description
userId	Y	(string) User id

txId	Y	(long) Transaction id of saba system.
refId	Y	(string) Unique id.
updateTime	Y	Update time when bet has changed. (yyyy-MM-dd HH:mm:ss.SSS) GMT-4
winlostDate	Y	The time in which the win lose is finalized(show date only). (yyyy-MM-dd 00:00:00.000) GMT-4
status	Y	(string)Transaction half won/half lose/won/lose/draw/void/refund/reject
payout	Y	(decimal) The winning amount for ticket.
creditAmount	Y	(decimal) The amount should be credited on player.
debitAmount	Y	(decimal) The amount should be deducted from player.
extraStatus	N	(String) Will have value in the following cases. -Ticket has been cash out-> "extraStatus": "cashout". In this case will send via Settle API. -Ticket has been voided after ticket cash out -> "extraStatus": "void ticket". In this case will send via Resettle API. -Match has been refunded after ticket cash out -> "extraStatus": "refund match". In this case will send via Resettle API. -Ticket has been voided cash out action after ticket cash out -> "extraStatus": "void cashout". In this case will send via Unsettle API.
extraInfo	Y	Reference to table of extraInfo.
settlementTime	Y	The time in which the ticket is settled. If ticket status=reject ,void, settlementTime will be "null".
odds	N	(decimal) This parameter will show only if bettype=468,469.

ExtraInfo

Field	Mandatory	Description
isOnlyWinlostDate Changed	Y	(boolean) The only change is WinlostDate or not and all other parameters such as status, creditAmount, debitAmount, or payout remain unchanged. e.g. true, false

For example:

```
{
  "key": "xxx",
  "message": {
    "action": "Resettle",
    "operationId": "4200500_1_25",
    "txns": [
      {
        "userId": "player01",
        "refId": "4200500_555",
        "txId": 339482738748,
        "updateTime": "2021-09-08T04:49:32.383-04:00",
        "winlostDate": "2021-09-08T00:00:00.000-04:00",
        "status": "won",
        "payout": 3.5600,
        "creditAmount": 3.5600,
        "debitAmount": 0.0,
        "extraStatus": "",
        "settlementTime": "2021-09-08T04:49:26.3",
        "extraInfo": {
          "isOnlyWinlostDateChanged": false
        }
      }
    ]
  }
}
```

```

    ]
  }
}

```

- Response Parameter

Field	Mandatory	Type	Description
status	Y	String	status code (Appendix: Status Code)
msg	N	string	If status is failed, then return description for this failure.

Successful Result

```

{
  "status": "0"
}

```

Failed Result

```

{
  "status": "203",
  "msg": "Account is not exist"
}

```

7. Unsettle

- Description

- ✧ Game provider is able to call this function that will be provided by merchant.
- ✧ Will repeatedly call Unsettle until it is successfully implemented or reach the limit in retry progress if it is failed to call in the first attempt. Can refer Appendix: Retry Mechanism for further detail.
- ✧ **The operationId is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.**

- Method URL

Provide by Merchant.

For Example: https://<Operator_API_SERVICE_URL>/unsettle

- Request Parameter

Field	Type	Mandatory	Description
key	string	Y	Value of key in credential document.
message	json	Y	Json Format: reference to table below

JSON Information:

Field	Mandatory	Description
action	Y	(string) Unsettle
operationId	Y	(string) It is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.
txns	Y	Json Format: reference to table below

****If there are multiple transactions, the transaction information can be sent as an array**

Field	Mandatory	Description
userId	Y	(string) user id
txId	Y	(long) Transaction id of saba system.
refId	Y	(string) Unique id.

updateTime	Y	Update time when bet has changed. (yyyy-MM-dd HH:mm:ss.SSS) GMT-4
creditAmount	Y	(decimal) The amount should be credited on player.
debitAmount	Y	(decimal) The amount should be deducted from player.
extraStatus	N	(String) Will have value in the following cases. -Ticket has been cash out-> "extraStatus": "cashout". In this case will send via Settle API. -Ticket has been voided after ticket cash out -> "extraStatus": "void ticket". In this case will send via Resettle API. -Match has been refunded after ticket cash out -> "extraStatus": "refund match". In this case will send via Resettle API. -Ticket has been voided cash out action after ticket cash out -> "extraStatus": "void cashout". In this case will send via Unsettle API.

For example:

```
{
  "key": "xxx",
  "message": {
    "action": "Unsettle",
    "operationId": "4200500_1_25",
    "txns": [
      {
        "userId": "player01",
        "refId": "4200500_555",
        "txId": 339482738748,
        "updateTime": "2021-09-08T04:49:32.383-04:00",
        "creditAmount": 0.0,

```

```

        "debitAmount":3.5600,

        "extraStatus": ""

    }

]

}

}

```

- Response Parameter

Field	Mandatory	Type	Description
status	Y	String	status code (Appendix: Status Code)
msg	N	string	If status is failed, then return description for this failure.

Successful Result

```

{

    "status": "0"

}

```

Failed Result

```

{

    "status":"203",

    "msg":"Account is not exist"

}

```

8. Place Bet Parlay

- Description

Game provider able to call this function that will be provided by merchant , and send out the bet details to merchant.

- ✧ Whenever the failed to place bet on Saba system, game provider will call the Cancel Bet API to cancel the reserve bet.
- ✧ If doesn't receive response, game provider will also call the Cancel Bet API to cancel the reserve bet.
- ✧ Only support decimal odds.
- ✧ **The operationId is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.**

- Method URL

[Provide by Merchant.](#)

For Example: https://<Operator_API_SERVICE_URL>/placebetparlay

- Request Parameter

Field	Type	Mandatory	Description
key	string	Y	Value of key in credential document
message	json	Y	Json Format: reference to table below

JSON Information:

Field	Mandatory	Description
action	Y	(string) PlaceBetParlay
operationId	Y	(string) It is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.
userId	Y	(string) Bet user id

currency	Y	(int) OW Currency Id e.g. 1, 2, 20 Please refer to Appendix: Currency Table
betTime	Y	(string) Place bet time (yyyy-MM-dd HH:mm:ss.SSS) GMT-4
updateTime	Y	(string) Update time when bet has changed (yyyy-MM-dd HH:mm:ss.SSS) GMT-4
totalBetAmount	Y	(decimal) Total bet amount (for check balance)
IP	Y	(string) e.g. 61.221.35.49 (IPv4)
tsId	N	(string) optional, user login session id sent by merchant
betFrom	Y	(string) The platform in which the bet came from. Please refer to Appendix: BetFrom Table
creditAmount	Y	(decimal) The total amount should be credited on player.
debitAmount	Y	(decimal) The total amount should be deducted from player.
vendorTransId	N	(string) This parameter will be returned only if the bet is placed through Direct API and Odds Feed API.
txns	Y	(array) Reference to table of Combo Info
ticketDetail	Y	(array) Reference to table of Ticket Detail .

Combo Info

refId	Y	(string) Unique id.
parlayType	Y	(string) e.g. Parlay_Mix, Parlay_System, Parlay_Lucky, SingleBet_ViaLucky
betAmount	Y	(decimal) Bet amount of ticket.
creditAmount	Y	(decimal) The total amount should be credited on player.
debitAmount	Y	(decimal) The total amount should be deducted from player.
detail	N	(array) SingleBet_ViaLucky -- reference to table of Single Bet Detail . Others -- reference to table of Parlay Detail .

Parlay Detail

type	Y	(int) e.g. 0, 1, 2 (Appendix: Combo Type)
name	Y	(string) e.g. Treble (1 Bet), Trixie (4 Bets)

		(Appendix: Combo Type)
betCount	Y	(int) e.g. 1, 3, 4
stake	Y	(decimal) Input bet amount
odds	N	(decimal) Show in Parlay_Mix only.

Single Bet Detail

matchId	Y	(int) e.g. 35627959
---------	---	---------------------

Ticket Detail

matchId	Y	(int) e.g. 35627959
homeId	Y	(int) e.g. 23, 24
awayId	Y	(int) e.g. 23, 24
homeName	Y	(string) This value depends on player's language. e.g. Chile (V)
awayName	Y	(string) This value depends on player's language. e.g. France (V)
kickOffTime	Y	(string) Kick off time (yyyy-MM-dd HH:mm:ss.SSS) GMT-4
sportType	Y	(int) e.g. 1, 2, 3 Please refer to Appendix: Sports Table
sportTypeName	Y	(string) This value depends on player's language. e.g. Soccer
betType	Y	(int) e.g. 1, 3 Please refer to Appendix: BetType Table
betTypeName	Y	(string) This value depends on player's language. e.g. Handicap
oddsId	Y	(int) e.g. 246903111
odds	Y	(decimal) e.g. -0.95, 0.75
oddsType	Y	(short) e.g. 1, 2, 3, 4, 5 Please refer to Appendix: OddsType Table
betChoice	Y	(string) e.g. Over, 4-3
betChoice_en	Y	(string) BetChoice in English. e.g. Over, 4-3
leagueId	Y	(int) e.g. 152765
leagueName	Y	(string) This value depends on player's language .e.g. SABA INTERNATIONAL FRIENDLY Virtual PES 20 - 20 Mins Play
isLive	Y	(boolean) e.g. true, false

point	N	(string) Line.
point2	N	(string) Line2. Only bettype 646 with value, Point is HDP and Point2 is OU.
betTeam	Y	(string)The team/side that customer placed bet on
homeScore	Y	(int) The home team's score when placing bet
awayScore	Y	(int) The away team's score when placing bet
baStatus	Y	(boolean)Flag of Bet Aggregator true/false
excluding	N	(string)Only returned when bet_team=aos. The returned value means that the correct score player placed on is not the listed options
leagueName_en	Y	(string)League name in English. e.g. JAPAN J-LEAGUE DIVISION 2
sportTypeName_en	Y	(String) Sport type name in English. e.g. Soccer
homeName_en	Y	(String) Home name in English. e.g. Chile (V)
awayName_en	Y	(String) Away name in English. e.g. France (V)
betTypeName_en	Y	(String) Bet type name in English. e.g. Handicap
matchDateTime	Y	(string) Match date time (yyyy-MM-dd HH:mm:ss.SSS) GMT-4 Note: matchDatetime of Outright Betting will be KickOffTime.
betRemark	N	(string) bettype 9662~9667(Cricket), 9676~9705(Cricket) with value. Bet type: 9662~9667 Will display two players name Bet type: 9676~9705 Will display one player name

For example:

```
{
  "key": "xxx",
  "message": {
    "action": "PlaceBetParlay",
    "operationId": "4200500_1_10832",
    "userId": "player",
    "currency": 20,
    "betTime": "2021-10-14T02:21:38.436-04:00",
```

```

"updateTime": "2021-10-14T02:21:38.436-04:00",
"totalBetAmount": 31,
"IP": "61.221.35.49",
"tsId": "12kdcel3#8772",
"betFrom": "x",
"creditAmount": 0,
"debitAmount": 31,
"vendorTransId": "test20230427001",
"txns": [
  {
    "refId": "4200500_2235",
    "parlayType": "Parlay_Mix",
    "betAmount": 3,
    "creditAmount": 0,
    "debitAmount": 3,
    "detail": [
      {
        "type": 1,
        "name": "Trebles",
        "betCount": 1,
        "stake": 3,
        "odds": 7.4216
      }
    ]
  },
  {
    "refId": "4200500_2236",
    "parlayType": "Parlay_Lucky",
    "betAmount": 16,
    "creditAmount": 0,
    "debitAmount": 16,

```

```

    "detail": [
      {
        "type": 3,
        "name": "Lucky7",
        "betCount": 4,
        "stake": 4
      }
    ]
  },
  {
    "refId": "4200500_2237",
    "parlayType": "SingleBet_ViaLucky",
    "betAmount": 4,
    "creditAmount": 0,
    "debitAmount": 4,
    "detail": [
      {
        "matchId": 46875647
      }
    ]
  },
  {
    "refId": "4200500_2238",
    "parlayType": "SingleBet_ViaLucky",
    "betAmount": 4,
    "creditAmount": 0,
    "debitAmount": 4,
    "detail": [
      {
        "matchId": 46875655
      }
    ]
  }

```

```

    ]
  },
  {
    "refId": "4200500_2239",
    "parlayType": "SingleBet_ViaLucky",
    "betAmount": 4,
    "creditAmount": 0,
    "debitAmount": 4,
    "detail": [
      {
        "matchId": 46875653
      }
    ]
  }
],
"ticketDetail": [
  {
    "matchId": 46875647,
    "homeId": 30588,
    "awayId": 452,
    "homeName": "Giravanz Kitakyushu",
    "awayName": "Tokyo Verdy",
    "kickOffTime": "2021-10-16T00:59:59.000-04:00",
    "sportType": 1,
    "sportTypeName": "Soccer",
    "betType": 1,
    "betTypeName": "Handicap",
    "oddsId": 370758567,
    "odds": 1.74,
    "oddsType": 3,
    "betChoice": "Tokyo Verdy",

```

```

    "betChoice_en": "Tokyo Verdy",
    "leagueId": 344,
    "leagueName": "JAPAN J-LEAGUE DIVISION 2",
    "isLive": false,
    "point": "0.00",
    "point2": "",
    "betTeam": "a",
    "homeScore": 0,
    "awayScore": 0,
    "baStatus": false,
    "excluding": "",
    "leagueName_en": "JAPAN J-LEAGUE DIVISION 2",
    "sportTypeName_en": "Soccer",
    "homeName_en": "Giravanz Kitakyushu",
    "awayName_en": "Tokyo Verdy",
    "betTypeName_en": "Handicap",
    "matchDateTime": "2021-10-16T01:00:00.000-04:00"
    "betRemark": ""
  },
  {
    "matchId": 46875655,
    "homeId": 205318,
    "awayId": 5833,
    "homeName": "SC Sagamihara",
    "awayName": "Montedio Yamagata",
    "kickOffTime": "2021-10-16T00:59:59.000-04:00",
    "sportType": 1,
    "sportTypeName": "Soccer",
    "betType": 3,
    "betTypeName": "Over/Under",
    "oddsId": 370758607,

```

```

    "odds": 2.21,
    "oddsType": 3,
    "betChoice": "Under",
    "betChoice_en": "Under",
    "leagueId": 344,
    "leagueName": "JAPAN J-LEAGUE DIVISION 2",
    "isLive": false,
    "point": "2.00",
    "point2": "",
    "betTeam": "a",
    "homeScore": 0,
    "awayScore": 0,
    "baStatus": false,
    "excluding": "",
    "leagueName_en": "JAPAN J-LEAGUE DIVISION 2",
    "sportTypeName_en": "Soccer",
    "homeName_en": "SC Sagamihara",
    "awayName_en": "Montedio Yamagata",
    "betTypeName_en": "Over/Under",
    "matchDateTime": "2021-10-16T01:00:00.000-04:00"
    "betRemark": ""
  },
  {
    "matchId": 46875653,
    "homeId": 44548,
    "awayId": 1825,
    "homeName": "V-Varen Nagasaki",
    "awayName": "Albirex Niigata",
    "kickOffTime": "2021-10-16T00:59:59.000-04:00",
    "sportType": 1,
    "sportTypeName": "Soccer",

```

```

        "betType": 3,
        "betTypeName": "Over/Under",
        "oddsId": 369196143,
        "odds": 1.93,
        "oddsType": 3,
        "betChoice": "Over",
        "betChoice_en": "Over",
        "leagueId": 344,
        "leagueName": "JAPAN J-LEAGUE DIVISION 2",
        "isLive": false,
        "point": "2.25",
        "point2": "",
        "betTeam": "h",
        "homeScore": 0,
        "awayScore": 0,
        "baStatus": false,
        "excluding": "",
        "leagueName_en": "JAPAN J-LEAGUE DIVISION 2",
        "sportTypeName_en": "Soccer",
        "homeName_en": "V-Varen Nagasaki",
        "awayName_en": "Albirex Niigata",
        "betTypeName_en": "Over/Under",
        "matchDateTime": "2021-10-16T01:00:00.000-04:00"
        "betRemark": ""
    }
}
}

```

- Response Parameter

Field	Mandatory	Type	Description
status	Y	string	Status code (Appendix: Status Code)
txns	Y	array	Reference to table of <i>TicketInfoMapping</i>
msg	N	string	If status is failed, then return description for this failure.

TicketInfoMapping

Field	Mandatory	Type	Description
refId	Y	string	Get from the parameters of request.
licenseeTxId	Y	string	Transaction id of merchant system.

Successful Result

```
{
  "status": 0,
  "txns":[
    {
      "refId":"4200500_2235",
      "licenseeTxId":"2453445"
    },
    {
      "refId":"4200500_2236",
      "licenseeTxId":"2453456"
    },
    {
```

```
"refId":"4200500_2237",  
  
"licenseTxId":"2453463"  
  
},  
  
{  
  
"refId":"4200500_2238",  
  
"licenseTxId":"2453511"  
  
},  
  
{  
  
"refId":"4200500_2239",  
  
"licenseTxId":"2453532"  
  
}}  
  
}
```

Failed Result

```
{  
  
"status":"203",  
  
"msg":"Account is not exist"  
  
}
```

9. Confirm Bet Parlay

● Description

Game provider is able to call this function that provided by merchant.

- ✧ When PlaceBetParlay is successful on provider side, game provider will call the ConfirmBetParlay API.
- ✧ Will repeatedly call Confirm Bet Parlay until it is successfully implemented or reach the limit in retry progress if it is failed to call in the first attempt. Can refer Appendix: Retry Mechanism for further detail.
- ✧ When the odds is changed to better one, request.isOddsChanged should be true then request.odds and request.actualAmount will be different from PlaceBetParlay.
- ✧ **The operationId is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.**

● Method URL

Provide by Merchant.

For Example: https://<Operator_API_SERVICE_URL>/confirmbetparlay

● Request Parameter

Field	Type	Mandatory	Description
key	string	Y	Value of key in credential document.
message	json	Y	Json Format: reference to table below.

JSON Information:

Field	Mandatory	Description
action	Y	(string) ConfirmBetParlay
operationId	Y	(string) It is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.
userId	Y	(string) user id
updateTime	Y	(string) Update time when bet has changed. (yyyy-MM-dd HH:mm:ss.SSS) GMT-4

creditAmount	Y	(decimal) The amount should be credited on player.
debitAmount	Y	(decimal) The amount should be deducted from player.
txns	Y	(array) Reference to table of <i>Ticket Info.</i>
ticketDetail	Y	(array) Reference to table of <i>Ticket Detail.</i>
transactionTime	Y	(string) The transaction time in which bet was placed in Saba system. (yyyy-MM-dd HH:mm:ss.SSS) GMT-4

Ticket Info:

Field	Mandatory	Description
refId	Y	(string) Unique id.
txId	Y	(long) Transaction id of saba system.
licenseeTxId	Y	(string) Transaction id of merchant system.
actualAmount	Y	(decimal) Actual bet amount
isOddsChanged	Y	(boolean) e.g. true, false
creditAmount	Y	(decimal) The amount should be credited on player.
debitAmount	Y	(decimal) The amount should be deducted from player.
winlostDate	Y	(string) The time in which the win lose is finalized(show date only). (yyyy-MM-dd 00:00:00.000) GMT-4
odds	N	(decimal) Show in Parlay_Mix only.

Ticket Detail:

Field	Mandatory	Description
matchId	Y	(int) e.g. 35627959
sportType	Y	(int) e.g. 1, 2, 3 Please refer to Appendix: Sports Table
betType	Y	(int) e.g. 1, 3 Appendix: BetType Table
oddsId	Y	(int) e.g. 246903111
odds	Y	(decimal) e.g. -0.95, 0.75

oddsType	Y	(short) e.g. 1, 2, 3, 4, 5 Please refer to Appendix: OddsType Table
leagueld	Y	(int) e.g. 152765
isLive	Y	(boolean) e.g. true, false
isOddsChanged	Y	(boolean) e.g. true, false

For example:

```
{
  "key": "xxx",
  "message": {
    "action": "ConfirmBetParlay",
    "userId": "player",
    "operationId": "4200500_2_10838",
    "updateTime": "2021-10-14T02:21:38.624-04:00",
    "creditAmount": 0,
    "debitAmount": 0,
    "transactionTime": "2021-10-14T02:21:38.000-04:00",
    "txns": [
      {
        "refId": "4200500_2235",
        "txId": 113545813706,
        "licenseeTxId": "2235",
        "actualAmount": 3,
        "isOddsChanged": false,
        "creditAmount": 0,
        "debitAmount": 0,
        "winlostDate": "2021-10-16T00:00:00.000-04:00",
        "odds": 7.4216
      },
      {
        "refId": "4200500_2237",
        "txId": 113545813703,
```

```

    "licenseeTxId": "2237",
    "actualAmount": 4,
    "isOddsChanged": false,
    "creditAmount": 0,
    "debitAmount": 0,
    "winlostDate": "2021-10-16T00:00:00.000-04:00"
  },
  {
    "refId": "4200500_2238",
    "txId": 113545813704,
    "licenseeTxId": "2238",
    "actualAmount": 4,
    "isOddsChanged": false,
    "creditAmount": 0,
    "debitAmount": 0,
    "winlostDate": "2021-10-16T00:00:00.000-04:00"
  },
  {
    "refId": "4200500_2239",
    "txId": 113545813705,
    "licenseeTxId": "2239",
    "actualAmount": 4,
    "isOddsChanged": false,
    "creditAmount": 0,
    "debitAmount": 0,
    "winlostDate": "2021-10-16T00:00:00.000-04:00"
  },
  {
    "refId": "4200500_2236",
    "txId": 113545813707,
    "licenseeTxId": "2236",

```

```

        "actualAmount": 16,
        "isOddsChanged": false,
        "creditAmount": 0,
        "debitAmount": 0,
        "winlostDate": "2021-10-16T00:00:00.000-04:00"
    }
],
"ticketDetail": [
    {
        "matchId": 46875647,
        "sportType": 1,
        "betType": 1,
        "oddsId": 370758567,
        "odds": 1.74,
        "oddsType": 3,
        "leagueId": 344,
        "isLive": false,
        "isOddsChanged": false
    },
    {
        "matchId": 46875655,
        "sportType": 1,
        "betType": 3,
        "oddsId": 370758607,
        "odds": 2.21,
        "oddsType": 3,
        "leagueId": 344,
        "isLive": false,
        "isOddsChanged": false
    },
    {

```

```

        "matchId": 46875653,
        "sportType": 1,
        "betType": 3,
        "oddsId": 369196143,
        "odds": 1.93,
        "oddsType": 3,
        "leagueId": 344,
        "isLive": false,
        "isOddsChanged": false
    }
}
}
}

```

- Response Parameter

Field	Mandatory	Type	Description
status	Y	string	status code (Appendix: Status Code)
balance	Y	decimal	If status is success, return balance for update. 2 decimals
msg	N	string	If status is failed, then return description for this failure.

Successful Result

```

{
    "status": "0",
    "balance": 100.00
}

```

Failed Result


```
{  
  "status": "203",  
  "msg": "Account is not exist"  
}
```

10. Place Bet 3rd

- Description

Game provider is able to call this function that will be provided by merchant, and send out the bet details to merchant.

- ✧ This method supports RNG Keno, Lottery, Table Game.
- ✧ Whenever the failed to place bet on Saba system, game provider will call the Cancel Bet API to cancel the reserve bet.
- ✧ If doesn't receive response, game provider will also call the Cancel Bet API to cancel the reserve bet.
- ✧ **The operationId is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.**

- Method URL

Provide by Merchant.

For Example: https://<Operator_API_SERVICE_URL>/placebet3rd

- Request Parameter

Field	Type	Mandatory	Description
key	string	Y	Value of key in credential document.
message	json	Y	Json Format: reference to table below.

JSON Information:

Field	Mandatory	Description
action	Y	(string) PlaceBet3rd
operationId	Y	(string) It is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.
userId	Y	(string) User id
currency	Y	(int) Saba Currency Id e.g. 1, 2, 20 Please refer to Appendix: Currency Table
productId	Y	(int) Sport type

gameId	Y	(int) Bet type
ticketList	Y	(array) Reference to table of <i>TicketDetail</i>
betTime	Y	(string) Place bet time (yyyy-MM-dd HH:mm:ss.SSS) GMT-4
IP	Y	(string) IPV4 or IPV6. e.g. 61.221.35.49 (IPV4)
tsId	N	(string) optional, user login session id sent by merchant
productName_en	Y	(string)Product name(sport type) in English.
gameName_en	Y	(string)Game name(bet type) in English.
betFrom	Y	(string) The platform in which the bet came from. Please refer to Appendix: BetFrom Table
creditAmount	Y	(decimal) The amount should be credited on player.
debitAmount	Y	(decimal) The amount should be deducted from player.

TicketDetail

refId	Y	(string) Unique id.
betChoice_en	Y	(string) BetChoice in English e.g. index small
odds	Y	(decimal) e.g. 0.53
oddsType	Y	(short) e.g. 1, 2, 3, 4, 5 Please refer to Appendix: OddsType Table
betAmount	Y	(decimal) Input bet amount
actualAmount	Y	(decimal) Actual bet amount
extra	N	Reference to table of <i>ExtraInfo</i> , it's different by productId(Sport type)
creditAmount	Y	(decimal) The amount should be credited on player.
debitAmount	Y	(decimal) The amount should be deducted from player.

ExtraInfo

RNG Keno (productId=202)

gameNo	Y	(int) Draw No.
range	Y	(int) Index range. e.g 5.
gameBetType	Y	(string) e.g.index big/small Please refer to Appendix: RNG Keno BetType Table

ExtraInfo

Lottery (productid=220), Table Game(productid=222)

gameNo	Y	(int) Draw No.
gameBetType	Y	(string) e.g. specific total Please refer to Appendix: Lottery Betttype Table & Table Game BetType Table

For example:

RNG Keno(productid=202)

```
{
  "key": "xxx",
  "message": {
    "action": "PlaceBet3rd",
    "operationId": "4200500_1_886746",
    "userId": "player",
    "currency": 20,
    "productId": 202,
    "gameId": 1528,
    "ticketList": [
      {
        "refId": "4200500_284432",
        "betChoice_en": "56",
        "odds": 75.0,
        "oddsType": 3,
        "betAmount": 12.0,
        "actualAmount": 12.0,
        "extra": {
          "gameNo": 1844037,
          "range": 0,
          "gameBetType": "special number"
        }
      }
    ]
  }
}
```

```

        },
        "creditAmount": 0.0,
        "debitAmount": 12.0
    },
    {
        "refId": "4200500_284433",
        "betChoice_en": "57",
        "odds": 75.0,
        "oddsType": 3,
        "betAmount": 12.0,
        "actualAmount": 12.0,
        "extra": {
            "gameNo": 1844037,
            "range": 0,
            "gameBetType": "special number"
        },
        "creditAmount": 0.0,
        "debitAmount": 12.0
    }
],
"betTime": "2023-10-03T05:10:15.704-04:00",
"IP": "61.221.35.49",
"tsId": "",
"productName_en": "RNG Keno",
"gameName_en": "Max Keno (20+1)",
"betFrom": "}",
"creditAmount": 0.0,
"debitAmount": 24.0
}
}

```

For example:

Lottery (productid=220), Table Game(productid=222)

```
{
  "key": "xxx",
  "message": {
    "action": "PlaceBet3rd",
    "operationId": "4200500_1_1174",
    "userId": "player2",
    "currency": 20,
    "productId": 220,
    "gameId": 1038,
    "ticketList": [
      {
        "refId": "4200500_411",
        "betChoice_en": "total(9)",
        "odds": 7,
        "oddsType": 3,
        "betAmount": 1,
        "actualAmount": 1,
        "extra": {
          "gameNo": 31835968,
          "gameBetType": "specific total"
        },
        "creditAmount": 0,
        "debitAmount": 1
      }
    ],
    "betTime": "2021-05-26T04:13:44.812-04:00",
    "IP": "61.221.35.49",
```

```

        "tsId": ""
        "productName_en": "Lottery",
        "gameName_en": "MaxDice",
        "betFrom": "x"
    }
}

```

● Response Parameter

Field	Mandatory	Type	Description
txns	Y	array	Reference to table of <i>TicketInfoMapping</i>
balance	Y	decimal	If status is success, return balance for update. 2 decimals
status	Y	string	Status code (Appendix: Status Code)
msg	N	string	If status is failed, then return description for this failure.

TicketInfoMapping

Field	Mandatory	Type	Description
refId	Y	string	Get from the parameters of request.
licenseeTxId	Y	string	Transaction id of merchant system.

Successful Result

```

{
    "txns": [
        {
            "refId": "4200500_284432",

```

```
        "licenseeTxId": "1277695"
      },
      {
        "refId": "4200500_284433",
        "licenseeTxId": "1277696"
      }
    ],
    "balance": 333.588,
    "status": "0"
  }
```

Failed Result

```
{
  "status": "203",
  "msg": "Account is not exist"
}
```


11. Confirm Bet 3rd

- Description

Game provider is able to call this function that provided by merchant.

- ✧ This method supports RNG Keno, Lottery, Table Game.
- ✧ When PlaceBet3rd is successful on provider side, game provider will call the ConfirmBet3rd API.
- ✧ Will repeatedly call Confirm Bet 3rd until it is successfully implemented or reach the limit in retry progress if it is failed to call in the first attempt. Can refer Appendix: Retry Mechanism for further detail.
- ✧ The operationId is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.

- Method URL

Provide by Merchant.

For Example: https://<Operator_API_SERVICE_URL>/confirmbet3rd

- Request Parameter

Field	Type	Mandatory	Description
key	string	Y	Value of key in credential document.
message	json	Y	Json Format: reference to table below.

JSON Information:

Field	Mandatory	Description
action	Y	(string) ConfirmBet3rd
userId	Y	(string) User id
operationId	Y	(string) It is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.
updateTime	Y	(string) (yyyy-MM-dd HH:mm:ss.SSS) GMT-4
txns	Y	(array) Reference to table of <i>TicketDetail</i>
transactionTime	Y	(string) The transaction time in which bet was placed in Saba system. (yyyy-MM-dd HH:mm:ss.SSS) GMT-4

creditAmount	Y	(decimal) The amount should be credited on player.
debitAmount	Y	(decimal) The amount should be deducted from player.

TicketDetail

refId	Y	(string) Unique id.
txId	Y	(long) Transaction id of saba system.
licenseeTxId	Y	(string) Transaction id of merchant system.
creditAmount	Y	(decimal) The amount should be credited on player.
debitAmount	Y	(decimal) The amount should be deducted from player.
winlostDate	Y	(string) The time in which the win lose is finalized(show date only). (yyyy-MM-dd 00:00:00.000) GMT-4

For example:

```
{
  "key": "",
  "message": {
    "action": "ConfirmBet3rd",
    "userId": "player",
    "operationId": "4200500_1_23",
    "updateTime": "2020-04-01T09:43:28.959-04:00",
    "transactionTime": "2020-04-01T09:44:28.959-04:00",
    "creditAmount": 0,
    "debitAmount": 0,
    "txns": [
      {
        "refId": "4200500_63",
        "txId": 849694463,
        "licenseeTxId": "5800",
        "creditAmount": 0,
        "debitAmount": 0,
        "winlostDate": "2020-04-02T00:00:00.000-04:00"
      }
    ]
  }
}
```

```

    }
  }
}

```

- Response Parameter

Field	Mandatory	Type	Description
status	Y	string	Status code (Appendix: Status Code)
msg	N	string	If status is failed, then return description for this failure.
balance	Y	decimal	If status is success, return balance for update. 2 decimals

Successful Result

```

{
  "status": "0",
  "msg": null,
  "balance": 1500
}

```

Failed Result

```

{
  "status": "203",

```

```
"msg": "Account is not exist"  
}
```

12. Place Bet Ent

- Description

Game provider is able to call this function that will be provided by merchant, and send out the bet details to merchant.

✧ This method supports Vgaming product.

- Method URL

[Provide by Merchant.](#)

For Example: https://<Operator_API_SERVICE_URL>/placebetent

- Request Parameter

Field	Type	Mandatory	Description
key	string	Y	Value of key in credential document.
message	json	Y	Json Format: reference to table below.

JSON Information:

Field	Mandatory	Description
action	Y	(string) PlaceBetEnt
userId	Y	(string) User id
currency	Y	(int) Saba Currency Id e.g. 1, 2, 20 Please refer to Appendix: Currency Table
productId	Y	(int) Sport type
gameId	Y	(int) Bet type
ticketList	Y	(array) Reference to table of <i>TicketDetail</i>
IP	Y	(string) IPV4 or IPV6 e.g. 61.221.35.49 (IPV4)
productName_en	Y	(string) Product name(sport type) in English.
gameName_en	Y	(string) Game name(bet type) in English.
betFrom	Y	(string) The platform in which the bet came from. Please refer to Appendix: BetFrom Table
roundId	Y	(string) Round number

betTime	Y	(string) Place bet time (yyyy-MM-dd HH:mm:ss.SSS) GMT-4
creditAmount	Y	(decimal) The amount should be credited on player.
debitAmount	Y	(decimal) The amount should be deducted from player.

TicketDetail

refId	Y	(string) Unique id. May receive multiple placebetent requests with same refId, which means that these requests are all placed on the same ticket.
betChoice_en	N	(string) BetChoice in English e.g. index small
stake	Y	(decimal) Input bet amount. If stake=0, it is Free Spin/Free Bet.
actualStake	Y	(decimal) Actual bet amount. This amount should be deducted from player. If actualStake=0, it is Free Spin/Free Bet.
extra	N	Reference to table of ExtraInfo , it's different by productId(Sport type)

ExtraInfo

gameNo	N	(int) Draw No.
range	N	(int) Index range. e.g 5.

For example:

```
{
  "key": xxx,
  "message": {
    "action": "PlaceBetEnt",
    "betTime": "2022-06-23T02:52:52.927-04:00",
    "userId": "Player",
    "ticketList": [
      {
        "refId": "171|$/64008486|$/3821|$/1165025",
        "stake": 10.0,
        "actualStake": 10.0
      }
    ]
  }
}
```

```

    }
  ],
  "currency": 20,
  "productId": 171,
  "productName_en": "Vgaming",
  "gameId": 3821,
  "gameName_en": "Sic Bo",
  "roundId": "1165025",
  "IP": "211.75.116.91",
  "betFrom": "x",
  "creditAmount": 0.0,
  "debitAmount": 10.0
}
}

```

- Response Parameter

Field	Mandatory	Type	Description
ticketList	Y	array	Reference to table of <i>TicketDetail</i>
userId	Y	string	User id
balance	Y	decimal	If status is success, return balance for update. 2 decimals
status	Y	string	Status code (Appendix: Status Code)
msg	N	string	If status is failed, then return description for this failure.

TicketDetail

Field	Mandatory	Type	Description
-------	-----------	------	-------------

refId	Y	string	Get from the parameters of request.
licenseeTxId	Y	string	Transaction id of merchant system.

Successful Result

```
{
  "userId": "Player",
  "balance": 100003.0120,
  "ticketList": [
    {
      "refId": "171|$/64008486|$/3821|$/1165025",
      "licenseeTxId": "410887"
    }
  ],
  "status": "0",
  "msg": "Success"
}
```

Failed Result

```
{
  "status": "203",
  "msg": "Account is not exist"
}
```


13. Settle Ent

- Description

Game provider is able to call this function that will be provided by merchant.

- ✧ This method supports Vgaming product.
- ✧ Game provider will send out the transaction after the game is complete.
- ✧ If there is resettlement, game provider will trigger "Get Ticket Info" API again and if the debit/credit amount need to be adjusted, we will send "Settle Ent" API with the same refId. And merchant has to modify balance accordingly.
- ✧ In normal case, ticket will be settled within 5 minutes. If not receive "Settle Ent" request, can call "Resubmit Process" API to check detail of this specific ticket.

- Method URL

Provide by Merchant.

For Example: https://<Operator_API_SERVICE_URL>/settleent

- Request Parameter

Field	Type	Mandatory	Description
key	string	Y	value of key in credential document
message	json	Y	Json Format: reference to table below.

JSON Information:

Field	Mandatory	Description
action	Y	(string) SettleEnt
userId	Y	(string) User id
refId	Y	(string) Unique id.
status	Y	(string) Ticket status. Vgaming (sporttype=171): won/lose/draw
actualStake	Y	(decimal) Actual deducted stake amount.
stake	Y	(decimal) Input bet amount.

netStake	Y	(decimal) Valid amount from game client. If you need to calculate the bonus/rebate, please base on this parameter.
winlostDate	Y	The time in which the win lose is finalized (show date only). (yyyy-MM-dd 00:00:00.000) GMT-4
creditAmount	Y	(decimal) The amount should be credited on player.
debitAmount	Y	(decimal) The amount should be deducted from player.
winlostAmount	Y	(decimal) Actual bet amount.
txIds	Y	(List<long>) Transaction ids of saba system. Will only have 1 txid on following product per request: Vgaming

For example:

```
{
  "key": xxx,
  "message": {
    "action": "SettleEnt",
    "userId": "Player",
    "refId": "171|$/64008486|$/3821|$/1165025",
    "status": "Lose",
    "actualStake": 50.0,
    "stake": 50.0,
    "netStake": 50.0,
    "winLostDate": "2022-06-23T00:00:00.000-04:00",
    "creditAmount": 40.0,
    "debitAmount": 0.0,
  }
}
```

```
        "winlostAmount": -10.0,

        "txIds": [

            104449148114

        ]

    }

}
```

- Response Parameter

Field	Mandatory	Type	Description
status	Y	string	status code (Appendix: Status Code)
msg	N	string	If status is failed, then return description for this failure.

Successful Result

```
{

    "status": "0"

}
```

Failed Result

```
{

    "status": "203",

    "msg": "Account is not exist"

}
```


14. Cancel Bet Ent

- Description

Game provider is able to call this function that will be provided by merchant.

- ✧ This method supports Vgaming product.
- ✧ When the player has placed bets and all have not been received by Saba & Vendor, and Merchant did not receive any settlement from Saba after placed bet 5 minutes later. Then Merchant will need to call API "Resubmit Process" to Saba to trigger the settlement process. Saba will return success. Then checks with the Vendor's status, if all the bets are not accepted successfully by vendor, then Saba calls API "Get Ticket Info" to get information to a calculation and cancel will be sent to Merchant.
- ✧ The cancellation only cancels the whole round, not to a specific bet.

- Method URL

Provide by Merchant.

For Example: https://<Operator_API_SERVICE_URL>/cancelbetent

- Request Parameter

Field	Type	Mandatory	Description
key	string	Y	Value of key in credential document.
message	json	Y	Json Format: reference to table below.

JSON Information:

Field	Mandatory	Description
action	Y	(string)CancelBetEnt
userId	Y	(string) User id
refId	Y	(string) Unique id.
txIds	Y	(List<long>) Transaction ids of saba system.
winlostDate	Y	The time in which the win lose is finalized (show date only). (yyyy-MM-dd 00:00:00.000) GMT-4
actualStake	Y	(decimal) Actual deducted stake amount.

winlostAmount	Y	(decimal) Actual bet amount.
creditAmount	Y	(decimal) The amount should be credited on player.
debitAmount	Y	(decimal) The amount should be deducted from player.

For example:

```
{
  "key": xxx,
  "message": {
    "action": "CancelBetEnt",
    "userId": "Player",
    "refId": "171|$/64008486|$/3821|$/1165025",
    "winLostDate": "2022-06-23T00:00:00.000-04:00",
    "actualStake": 40.0,
    "creditAmount": 40.0,
    "debitAmount": 0.0,
    "winlostAmount": -10.0,
    "txIds": [
      104449148114
    ]
  }
}
```

- Response Parameter

Field	Mandatory	Type	Description
status	Y	string	status code (Appendix: Status Code)

msg	N	string	If status is failed, then return description for this failure.
-----	---	--------	--

Successful Result

```
{  
  "status": "0"  
}
```

Failed Result

```
{  
  "status": "904",  
  "msg": " Request Timeout "  
}
```

15. Get Ticket Info

- Description

Game provider is able to call this function that will be provided by merchant.

- ✧ This method supports Vgaming product.
- ✧ Saba will need the necessary parameters from this API in order to calculate the correct credit & debit amount for settlement or cancellation.

- Method URL

Provide by Merchant.

For Example: https://<Operator_API_SERVICE_URL>/getticketinfo

- Request Parameter

Field	Type	Mandatory	Description
key	string	Y	Value of key in credential document.
message	json	Y	Json Format: reference to table below.

JSON Information:

Field	Mandatory	Description
action	Y	(string) GetTicketInfo
userId	Y	(string) User id
refId	Y	(string) Unique id.

For example:

```
{
  "key": xxx,
  "message": {
    "action": "GetTicketInfo",
    "userId": "Player",
```



```

      "refId": "171|$/64008486|$/3821|$/1165025"

    }

  }

```

- Response Parameter

Field	Mandatory	Type	Description
status	Y	string	status code (Appendix: Status Code)
msg	N	string	If status is failed, then return description for this failure.
ticketStatus	Y	String	Current status. Vgaming (sporttype=171) running/won/lose/draw/void/refund/reject
actualStake	Y	decimal	Actual deducted stake amount.
winlostAmount	Y	decimal	Actual bet amount.

Successful Result

```

{

  "status": "0",

  "msg": "Success",

  "ticketStatus": "running",

  "actualStake": 50.0,

  "winlostAmount": 0.0

}

```

Failed Result

```

{

```

```
"status": "904",  
  
"msg": " Request Timeout "  
  
}
```

16. Health Check

- Description

Game provider is able to call this function that will be provided by merchant.

✧ To Check if the connection between game provider and merchant is stable.

- Method URL

Provide by Merchant.

For Example: https://<Operator_API_SERVICE_URL>/healthcheck

- Request Parameter

Field	Type	Mandatory	Description
key	string	Y	Value of key in credential document.
message	json	Y	Json Format: reference to table below.

JSON Information:

Field	Mandatory	Description
action	Y	(string) HealthCheck
time	Y	(string) Time that HealthCheck be called.

For example:

```
{
  "key": " xxx ",
  "message":
    {
      "action": " HealthCheck ",
      " time ": "2021-11-09T05:29:33.325-04:00"
    }
}
```

```
}

```

- Response Parameter

Field	Mandatory	Type	Description
status	Y	string	status code (Appendix: Status Code)
msg	N	string	If status is failed, then return description for this failure.

Successful Result

```
{
  "status": "0"
}
```

Failed Result

```
{
  "status": "904",
  "msg": " Request Timeout "
}
```


17. Adjust Balance

- Description

Game provider is able to call this function that will be provided by merchant.

- ✧ This method supports promotions or any actions that will credit or debit player's balance.
- ✧ Will repeatedly call Adjust Balance until it is successfully implemented or reach the limit in retry progress if it is failed to call in the first attempt. Can refer Appendix: Retry Mechanism for further detail.
- ✧ **The operationId is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.**

- Method URL

Provide by Merchant.

For Example: https://<Operator_API_SERVICE_URL>/adjustbalance

- Request Parameter

Field	Type	Mandatory	Description
key	string	Y	Value of key in credential document.
message	json	Y	Json Format: reference to table below.

JSON Information:

Field	Mandatory	Description
action	Y	(string) AdjustBalance
time	Y	(string) Time that AdjustBalance be called.
userId	Y	(string) User id.
currency	Y	(int) Saba Currency Id e.g. 1, 2, 20 Please refer to Appendix: Currency Table
txId	Y	(long) Transaction id of saba system.
refNo	N	(long) Only bettype 17007 with have this parameter, and it's a parent ticket for risk free bet.
refId	Y	(string) Unique id.

operationId	Y	(string) It is a unique identifier. Please do not perform repeated credit or debit transactions on a player's wallet for requests with the same operationId.
betType	Y	(int) Will have following bet types. 17003 Saba Reward Cash Exchange 17007 Saba Reward Risk-free Rebate 17008 Saba Coins Cash Exchange 17006 Casino Royale Tournament Prize 17011 NBA Playoffs Lottery 17012 Balance Adjustment
betTypeName	Y	(string) Only English.
winlostDate	Y	(string) The time in which the win lose is finalized(show date only). (yyyy-MM-dd 00:00:00.000) GMT-4
balanceInfo	Y	Reference to table of BalanceInfo ,

BalanceInfo

creditAmount	Y	(decimal) The amount should be credited on player.
debitAmount	Y	(decimal) The amount should be deducted from player.

For example:

```
{
  "key": xxx,
  "message": {
    "action": "AdjustBalance",
    "time": "2022-07-01T06:44:13.717-04:00",
    "userId": "player01",
    "currency": 20,
```

```

        "txId": 789,

        "refNo": 0,

        "refId": "4282200_123_U",

        "operationId": "4282200_8_456_U",

        "betType": 17003,

        "betTypeName": "Saba Loyalty Points Exchange",

        "winlostDate": "2022-07-01T00:00:00.000-04:00",

        "balanceInfo": {

            "creditAmount": 10.0,

            "debitAmount": 0.0

        }

    }

}

```

● Response Parameter

Field	Mandatory	Type	Description
status	Y	string	status code (Appendix: Status Code)
msg	N	string	If status is failed, then return description for this failure.

Successful Result

```

{

    "status": "0"

}

```

Failed Result


```
{  
  "status": "904",  
  "msg": " Request Timeout "  
}
```

API provided by game provider

0. Integration Process

The following steps are recommended order to apply Single Wallet API.

First Step



To complete other assistant APIs

1. Check Ticket Status
2. Retry Operation
3. Get Reach Limit Trans
4. ResubmitProcess

Others



5. CheckUserOutstanding

1. Check Ticket Status

- Description

Merchant is able to call this function that will be provided by game provider.

- To check status of the specific ticket. This API provides the complete history of this specific ticket.
- Only can check the ticket within 100 days.

- Method URL

Same as Merchant API.

For Example: <https://<Merchant API URL>/api/checkticketstatus>

- Request Parameter

Field	Type	Mandatory	Description
vendor_id	string	Y	Value of key in credential document.
refId	string	Y	(string) Unique id.
txId	long	Y	(long) Transaction id of saba system.

- Either refId or txId needs to be sent.
- refId will take priority over txId if both refId and txId are sent.

For example:

```
vendor_id=xxx&refId=4200500_66&txId=849694463
```

- Response Parameter

Field	Type	Description
error_code	int	status code (Appendix: Status Code)
message	string	If status is failed, then return description for this failure.
betType	string	(int) e.g. 1, 3 Appendix: BetType Table

userId	string	(string)user id
licenseeTxId	string	Transaction id of merchant system.
winlostDate	string	(string) The time in which the win lose is finalized(show date only). (yyyy-MM-dd 00:00:00.000) GMT-4
txId	long	(long) Transaction id of saba system.
refId	string	(string) Unique id.
stake	decimal	(decimal) Bet amount
Odds	decimal	(decimal) e.g. -0.95, 0.75
transHistory	json	Json Format: reference to table below.

JSON Information:

Field	Type	Description
action	string	(string)Transaction Type e.g.PlaceBet, Settle
actionDate	string	(string)The id of transaction log.
stake	decimal	(decimal) Bet amount
odds	decimal	(decimal) e.g. -0.95, 0.75
status	short	(short) status 0=succeed , 1=failed, 2=pending.
winlostAmount	decimal	(decimal) The winning amount for ticket.
winlostDate	string	(string) The time in which the win lose is finalized(show date only). (yyyy-MM-dd 00:00:00.000) GMT-4
operationId	string	(string) The id of transaction log.
creditAmount	decimal	(decimal) The amount should be credited on player.
debitAmount	decimal	(decimal) The amount should be deducted from player.

inRetry	bool	(bool) If the retry mechanism is still in progress, then return true. Otherwise, it would be false.
reachRetryLimit	bool	(bool) If the retry mechanism had reached the limit, then return true. Otherwise, it would be false.

Successful Result

```
{
  "error_code": 0,
  "message": "",
  "Data": {
    "betType": 1038,
    "winlostDate": "2023-08-09T02:34:25.103",
    "txId": 150020050852511756,
    "userId": "player",
    "refId": "4200500_992852",
    "licenseeTxId": "3099880",
    "stake": 10.0,
    "odds": 30.0,
    "transHistory": [
      {
        "action": "PlaceBet",
        "actionDate": "2023-08-09T02:34:25.103",
        "stake": 10.0,
        "odds": 30.0,

```

```

        "status": 0,

        "winlostAmount": null,

        "winlostDate": null,

        "operationId": "4200500_1_3099880",

        "creditAmount": 0.0,

        "debitAmount": 10.0,

        "inRetry": false,

        "reachRetryLimit": false
    },
    {

        "action": "ConfirmBet",

        "actionDate": "2023-08-09T02:34:25.363",

        "stake": 10.0,

        "odds": 0.0,

        "status": 0,

        "winlostAmount": null,

        "winlostDate": null,

        "operationId": "4200500_2_3099881",

        "creditAmount": 0.0,

        "debitAmount": 0.0,

        "inRetry": false,

        "reachRetryLimit": false
    },
    {

```

```
        "action": "Settle",
        "actionDate": "2023-08-09T02:34:46.48",
        "stake": 10.0,
        "odds": 30.0,
        "status": 0,
        "winlostAmount": null,
        "winlostDate": "2023-08-09T00:00:00",
        "operationId": "4200500_4_3099887",
        "creditAmount": 0.0,
        "debitAmount": 0.0,
        "inRetry": false,
        "reachRetryLimit": false
    }
}
}
```

Failed Result

```
{
  "error_code": 101,
  "message": "ParameterIncorrect"
}
```

2. Retry Operation

- Description

Merchant is able to call this function that will be provided by game provider.

- To let the specific operation be resumed in retry progress.
- In order to proceed transaction properly. The API can be used for transaction retry of final status is failed/pending for specific operation.
- Placebet series cannot be retried. This API is not applicable for PlaceBet, PlaceBetParlay PlaceBet3rd and PlaceBetEnt.
- Only can check the ticket within 30 days for ConfirmBetParlay/ConfirmBet3rd, but others can check within 100 days
- Settle request will not be sent if ConfirmBet Request is not responded successfully within 7 days.
Please use api/RetryOperation to resend ConfirmBet and return success to us. Then, contact Saba technical window to manually process the settlement.

- Method URL

Same as Merchant API.

For Example: https://<Merchant_API_URL>/api/retryoperation

- Request Parameter

Field	Type	Mandatory	Description
vendor_id	string	Y	Value of key in credential document.
operationId	string	Y	(string) The id of transaction log.

For example:

```
vendor_id=xxx&operationId=4200500_4_387
```

- Response Parameter

Field	Type	Description
-------	------	-------------

error_code	int	status code (Appendix: Status Code)
message	string	If status is failed, then return description for this failure.

Successful Result

```
{  
  "error_code": 0,  
  "message": ""  
}
```

Failed Result

```
{  
  "error_code": 101,  
  "message": "ParameterIncorrect"  
}
```

3. Get Reach Limit Trans

- Description

Merchant is able to call this function that will be provided by game provider.

- To get all tickets which had reached the limit in retry progress.
- Only can get transactions within 30 days.
- It is recommended to implement a daily job in triggering "GetReachLimitTrans" to get the ticket list which has reached the maximum retry limit, and to further verify (CheckTicketStatus) and retry (RetryOperation) to ensure intact ticket process..

- Method URL

Provide by Merchant API.

For Example: https://<Merchant_API_SERVICE_URL>/api/getreachlimittrans

- Request Parameter

Field	Type	Mandatory	Description
vendor_id	string	Y	Value of key in credential document.
start_Time	String	Y	Time restriction within 30 days.

For example:

```
vendor_id=xxx&start_Time=2022/04/01
```

- Response Parameter

Field	Type	Description
txns	json	Json Format: reference to table below
error_code	int	status code (Appendix: Status Code)
message	string	If status is failed, then return description for this failure.

JSON Information:

Field	Type	Description
betType	int	e.g. 1, 3 Appendix: BetType Table
userId	string	user id
licenseeTxId	string	Transaction id of merchant system.
winlostDate	string	The time in which the win lose is finalized(show date only). (yyyy-MM-dd 00:00:00.000) GMT-4
txId	long	Transaction id of saba system.
refId	string	Unique id.
stake	decimal	Bet amount
odds	decimal	e.g. -0.95, 0.75
transHistory	json	Json Format: reference to table below.

JSON Information:

Field	Type	Description
action	string	Transaction Type e.g. PlaceBet, Settle
actionDate	string	The id of transaction log.
stake	decimal	Bet amount
odds	decimal	e.g. -0.95, 0.75

status	short	status 1=failed, 2=pending.
winlostAmount	decimal	The winning amount for ticket.
winlostDate	string	The time in which the win lose is finalized(show date only). (yyyy-MM-dd 00:00:00.000) GMT-4
operationID	string	The id of transaction log.
creditAmount	decimal	The amount should be credited on player.
debitAmount	decimal	The amount should be deducted from player.
inRetry	bool	If the retry mechanism is still in progress, then return true. Otherwise, it would be false.
reachRetryLimit	bool	If the retry mechanism had reached the limit, then return true. Otherwise, it would be false.

Successful Result

```

{
  "Txns": [
    {
      "betType": 1,
      "userId": "Player",
      "winlostDate": "2021-03-19T06:54:35.04",
      "licenseeTxId": "1988754",
      "txId": 12345678,
      "refId": "4200500_3",
      "stake": 10.0000,
      "odds": 1.3000,
    }
  ]
}
```

```

    "transHistory": [
      {
        "action": "Settle",
        "actionDate": "2021-03-19T06:56:18.937",
        "stake": 10.0000,
        "odds": 1.3000,
        "status": 1,
        "winlostAmount": 1.0000,
        "winlostDate": "2021-03-19T00:00:00",
        "operationId": "4200500_4_13",
        "creditAmount": 0.0,
        "debitAmount": 0.0,
        "inRetry": false,
        "reachRetryLimit": true
      }
    ],
    {
      "betType": 0,
      "userId": "Player",
      "winlostDate": "2021-05-18T05:49:33.067",
      "licenseeTxId": "19887549",
      "txId": 112233445566,
      "refId": "4200500_13",

```

```

        "stake": 100.0000,

        "odds": 1.5100,

        "transHistory": [

            {

                "action": "ConfirmBet",

                "actionDate": "2021-05-18T05:49:35.777",

                "stake": 100.0000,

                "odds": 1.5100,

                "status": 1,

                "operationId": "4200500_2_31",

                "creditAmount": 0.0,

                "debitAmount": 0.0,

                "inRetry": false,

                "reachRetryLimit": true

            }

        ],

    },

    {

        "betType": 1,

        "userId": "Player",

        "winlostDate": "2021-05-18T06:01:27.737",

        "licenseeTxId": "1988214",

        "txId": 112233445566,

        "refId": "4200500_16",

```

```

        "stake": 100.0000,

        "odds": 1.5100,

        "transHistory": [

            {

                "action": "ConfirmBet",

                "actionDate": "2021-05-18T06:01:28.13",

                "stake": 100.0000,

                "odds": 1.5100,

                "status": 1,

                "operationId": "4200500_2_37",

                "creditAmount": 0.0,

                "debitAmount": 0.0,

                "inRetry": false,

                "reachRetryLimit": true

            }

        ],

    },

    ],

    "error_code": 0,

    "message": ""

}

```

Failed Result

```
{
```

```
"error_code": 101,  
  
"message": "ParameterIncorrect"  
  
}
```


4. Resubmit Process

- Description

Merchant is able to call this function that will be provided by game provider.

- This method supports Vgaming product.
- This API can be requested while settlement is not received after 5 mins when a bet is placed. Then merchant will need to call this API "ResubmitProcess" to Saba to trigger the settlement process and Saba returns success when received the request. Then Saba will check with ticket status, if the status is running then no follow actions will be activated.
- Merchant will need to wait and call "ResubmitProcess" regularly or a settlement will be sent by Saba when the ticket status has returned from the vendor.
- The frequency can be set between 5 -10 mins interval as recommended.

- Method URL

Same as Merchant API.

For Example: https://<Merchant_API_URL>/api/resubmitprocess

- Request Parameter

Field	Type	Mandatory	Description
vendor_id	string	Y	Value of key in credential document.
refId	string	Y	(string) Unique id.
userId	String	Y	(string) User ID.

For example:

```
vendor_id=xxx&refId=175/$/14242435/$/3822/$/647384&userId=player
```

- Response Parameter

Field	Type	Description
-------	------	-------------

error_code	int	status code (Appendix: Status Code)
message	string	If status is failed, then return description for this failure.

Successful Result

```
{  
  "error_code": 0,  
  "message": ""  
}
```

Failed Result

```
{  
  "error_code": 101,  
  "message": "ParameterIncorrect"  
}
```

5. Check User Outstanding

- Description

Merchant is able to call this function that will be provided by game provider.

- This method supports to check users outstanding balance on game provider.

- Method URL

Same as Merchant API.

For Example: https://<Merchant_API_URL>/api/checkuseroutstanding

- Request Parameter

Field	Type	Mandatory	Description
vendorId	string	Y	Value of key in credential document.
vendorMemberIds	string	Y	(string) Vendor member id. Can input multiple users.

For example:

```
vendorId=xxx&vendorMemberIds=member01,member02
```

- Response Parameter

Field	Type	Description
errorCode	int	status code (Appendix: Status Code)
message	string	If status is failed, then return description for this failure.

Successful Result

```
{
```

```
"errorCode": 0,  
  
"message": "",  
  
"data": [  
  
  {  
  
    "vendorMemberId": "member01",  
  
    "outstanding": 0,  
  
    "currency": 20,  
  
    "errorCode": 0  
  
  },  
  
  {  
  
    "vendorMemberId": "member02",  
  
    "outstanding": 10,  
  
    "currency": 20,  
  
    "errorCode": 0  
  
  }  
  
]  
  
}
```

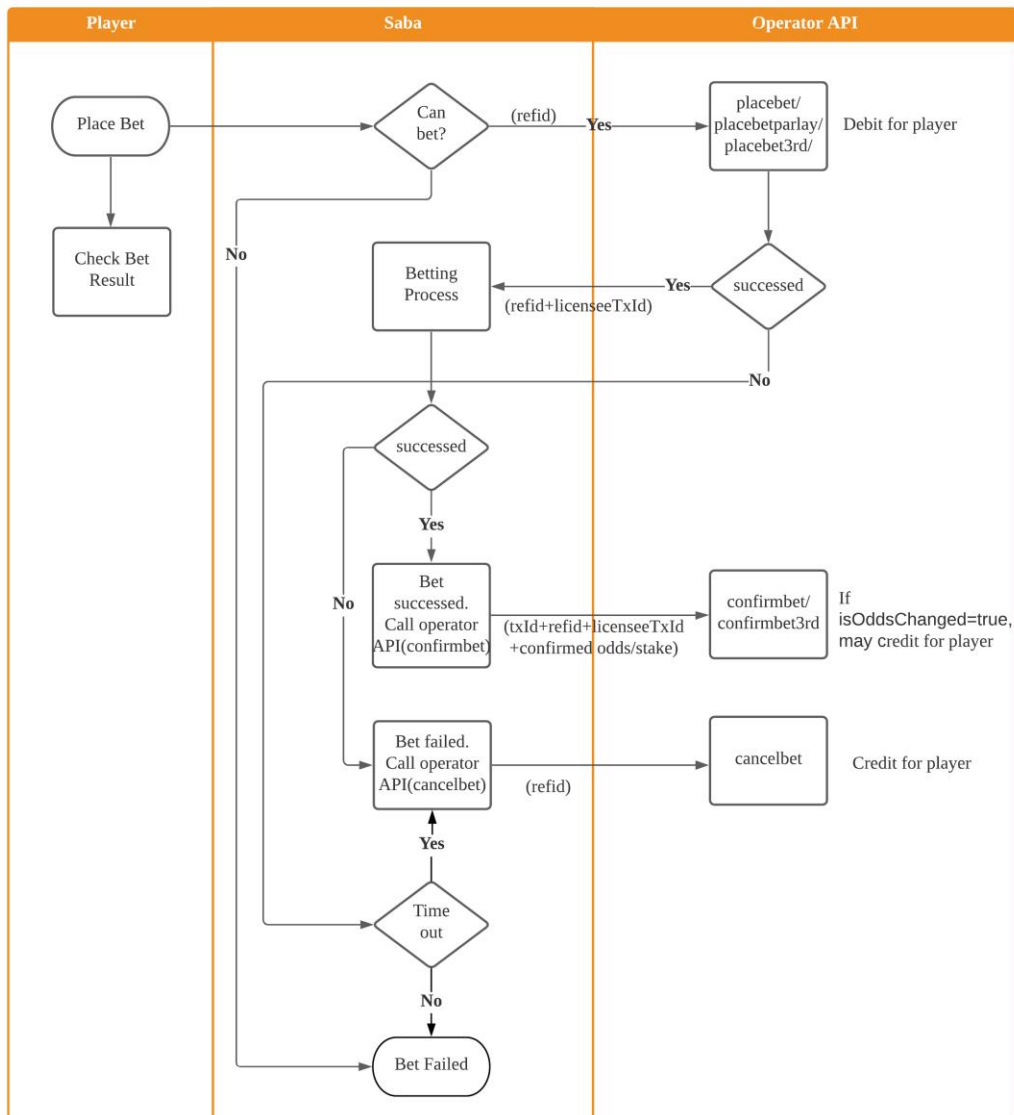
Failed Result

```
{  
  
  "errorCode": 101,  
  
  "message": "ParameterIncorrect"  
  
}
```

Appendix

1. Bet Flow

Bet flow of PlaceBet, PlaceBetParlay and PlaceBet3rd:

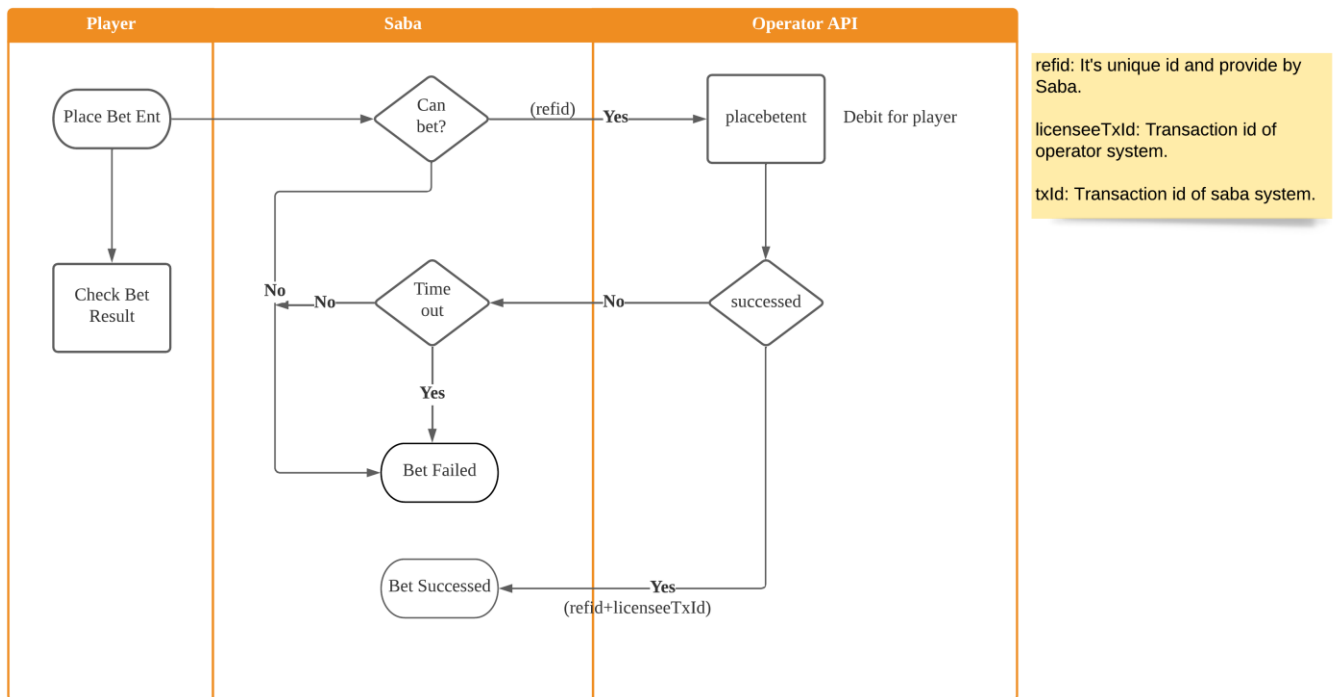


refid: It's unique id and provide by Saba.

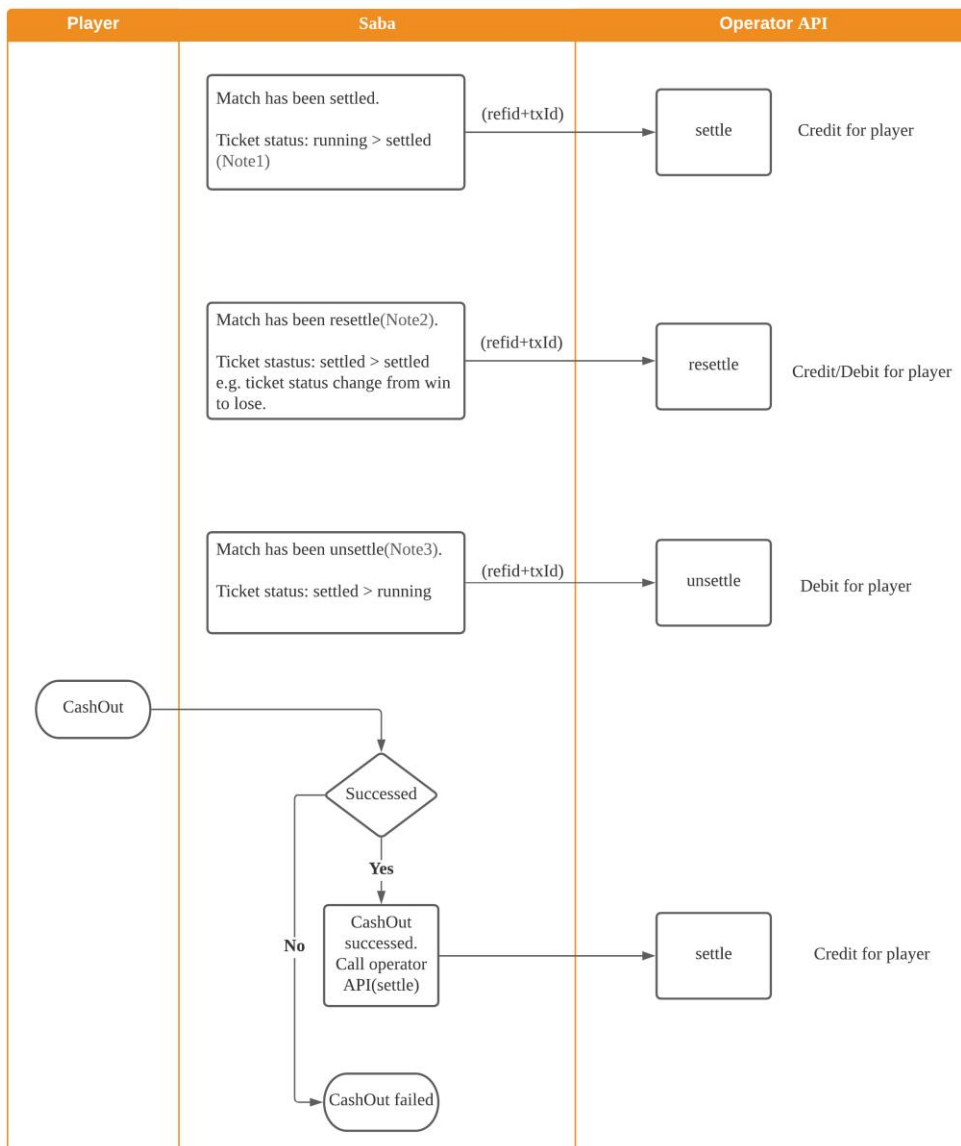
licenseeTxId: Transaction id of operator system.

txId: Transaction id of saba system.

Bet Flow of PlaceBetEnt:



2. Settlement Flow



Note 1:

Settled status includes win/ lose/ draw/ half lose/half win/ void/ reject/ refund.

Note 2:

1. Only partial tickets are settled due to system issue.
2. Human mistake.
3. Provider/scout mistake.
4. Match official make changes on certain judgement and change the outcome of the match in official websites.

Note 3:

1. Human mistake.
2. Provider/scout mistake but unable to give the confirmed outcome right away.

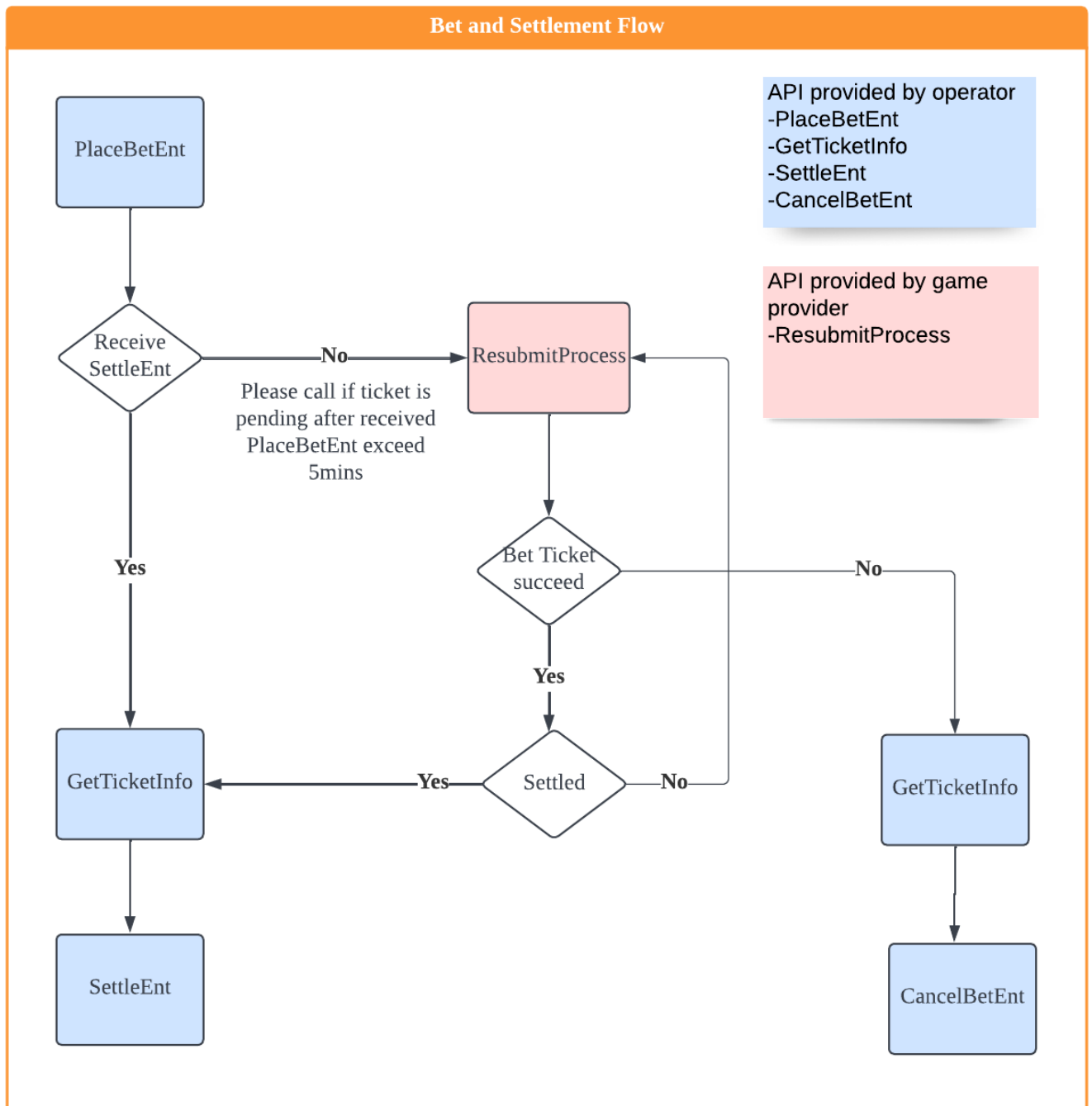
Remark1:

Ticket may occur more than 1 time settle/resettle/unsettle.

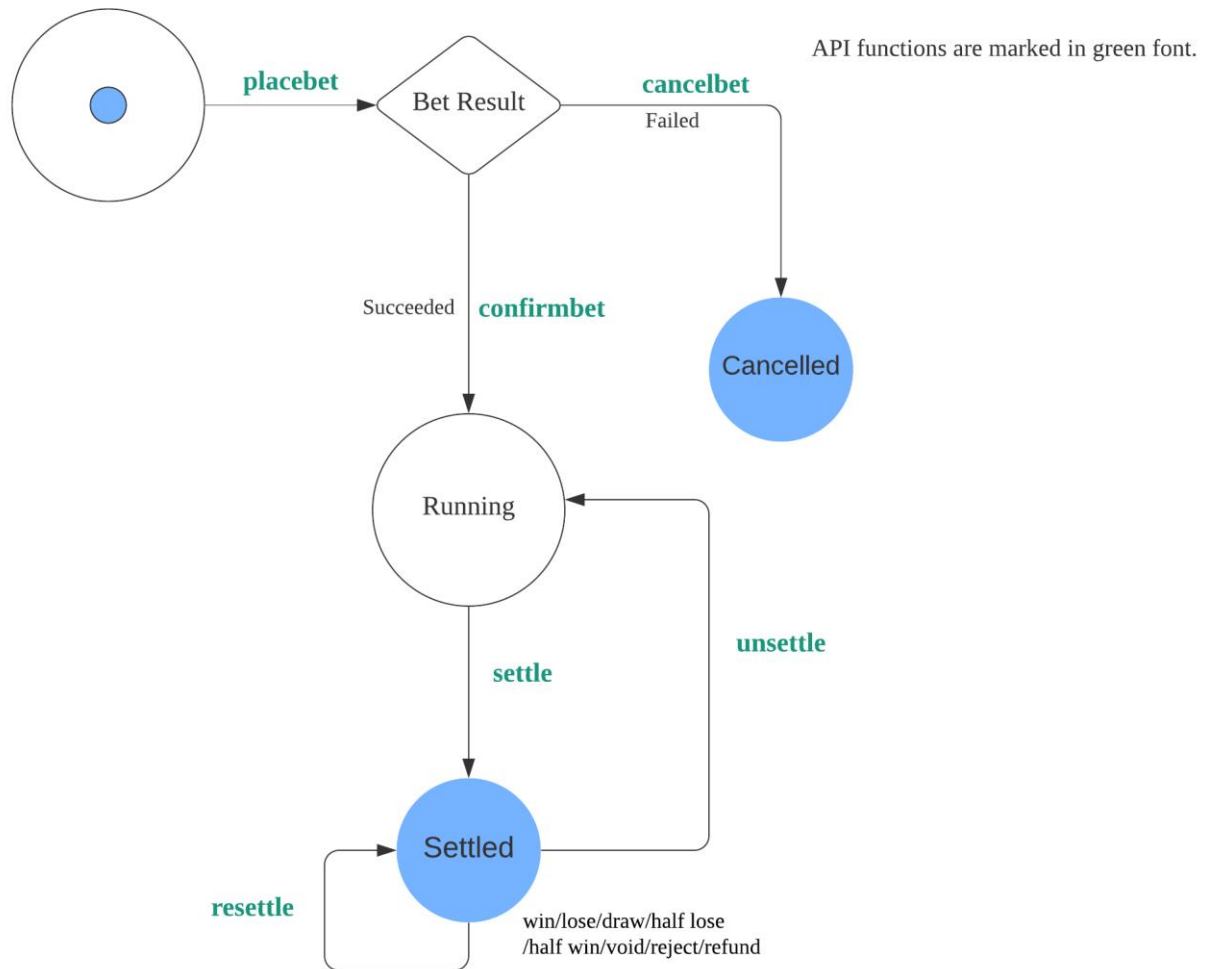
3. Ent Series Flow

-Bet and settlement flow of PlaceBetEnt, SettleEnt, CancelBetEnt, GetTicketInfo and ResubmitProcess.

-Support Vgaming product.



4. State Diagram



5. Supported Products & API list

	API provided by merchant					API provided by game provider	
	PlaceBet ConfirmBet	PlaceBetParlay ConfirmBetParlay	PlaceBet3rd ConfirmBet3rd	CancelBet Settle Resettle Unsettle	PlaceBetEnt SettleEnt CancelBetEnt GetTicketInfo	CheckTicketStatus RetryOperation GetReachLimitTrans	ResubmitProcess
Sportsbook SportType:1~99,177~199	✓	✓		✓		✓	
Number Game SportType:161,164	✓			✓		✓	
RNG KENO SportType:202,220,222			✓	✓		✓	
Vgaming SportType:171					✓		✓

6. Performance and Timeout

The table below lists invoked procedure and timeout mechanism of all the API functions.

Get Balance	
Functions	Timeout
Get Balance	30 Seconds

Health Check		
Functions	Invoke	Timeout
Health Check	10 Seconds	20 Seconds

Place Bet	
Functions	Timeout
Place Bet	3 Seconds
Place Bet Parlay	3 Seconds
Place Bet 3rd	3 Seconds
Place Bet Ent	3 Seconds

Confirm Bet	
Functions	Timeout
Confirm Bet	10 Seconds
Confirm Bet Parlay	10 Seconds
ConfirmBet3rd	10 Seconds

Cancel Bet	
Functions	Timeout
Cancel Bet	30 Seconds
Cancel Bet Ent	30 Seconds

Get Ticket Info	
Functions	Timeout
GetTicketInfo	30 Seconds

Settle	
Functions	Timeout
Settle	30 Seconds
Resettle	30 Seconds
Unsettle	30 Seconds
SettleEnt	30 Seconds

Adjust Balance	
Functions	Timeout
Adjust Balance	30 Seconds

7. Retry Mechanism

- **The following API functions do not have retry mechanism:**

- ✧ Place Bet
- ✧ Place Bet Parlay
- ✧ Place Bet 3rd
- ✧ Place Bet Ent
- ✧ Settle Ent
- ✧ Cancel Bet Ent
- ✧ Get Ticket Info

- **Invoke frequency and Times limit:**

- ✧ Retry will be invoked within every **30** seconds and with **100** times limit. (Cancel Bet Status Code: 504 will be retried 10 times; other Cancel Bet status codes will be retried 100 times.)
When the merchant system consistently experiences issues, it might lead to congestion and consequently affect the triggering frequency of this mechanism.

- **Specific error code**

- ✧ Receive the following error code listed below, retry mechanism will not be trigger.
- ✧ If receive the error code 903, the retry times won't be counted.
- ✧ If Cancel Bet receive the error code 504 will trigger the retry mechanism.

Functions	Error Code										
	1	203	302	303	306	307	308	309	504	505	903
All API (Except as follows)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	●
Cancel Bet	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	●

- ✧ ✓ : Will trigger the retry mechanism.
- ✧ ✗ : Will not trigger the retry mechanism.
- ✧ ● : Retry times won't be counted.

8. Status Code

Status Code (String)	Definition
-1	Undefined
0	Success
1	Duplicate Transaction
101	Parameter(s) Incorrect
201	Account Is Closed
202	Account Is Locked
203	Account Is Not Exist
204	Account Suspend
205	Account Existed
301	Merchant Id Is Not Exist
302	Invalid Currency
303	Invalid User Id
304	Language Is Not Support
305	Invalid Token
306	Invalid Time Zone
307	Invalid Amount
308	Invalid Date Time Format
309	Invalid Transaction Status
310	Invalid Bet Limit Setting
311	Invalid Authentication Key
312	Invalid IP Address
501	Permission Denied
502	Player Has Insufficient Funds
503	The IP address Is Restricted
504	No Such Ticket
505	No Data
506	Unable To Processed. Please Try Again Later
507	Bet Not Eligible For Bonus
508	Placebet series cannot be retried
901	Database Error
902	Network Error

903	System Is Under Maintenance
904	Request Timeout
905	System Busy
999	System Error

9. Combo Type

Combo Type	Definition
0	Doubles
1	Trebles
2	Trixie
3	Lucky7
4	Fold4
5	Yankee
6	Lucky15
7	Fold5
8	Canadian
9	Lucky31
10	Fold6
11	Heinz
12	Lucky63
13	Fold7
14	SuperHeinz
15	Lucky127
16	Fold8
17	Goliath
18	Lucky255
19	Fold9
20	Fold10
21	Fold11
22	Fold12
23	Fold13
24	Fold14
25	Fold15
26	Fold16
27	Fold17

28	Fold18
29	Fold19
30	Fold20
31	Fold21
32	Fold22
33	Fold23
34	Fold24
35	Fold25
36	Fold26
37	Fold27
38	Fold28
39	Fold29
40	Fold30
41	Fold31
42	Fold32
43	Fold33
44	Fold34
45	Fold35
46	Fold36
47	Fold37
48	Fold38
49	Fold39
50	Fold40
51	Fold41
52	Fold42
53	Fold43
54	Fold44
55	Fold45
56	Fold46
57	Fold47

58	Fold48
59	Fold49
60	Fold50
61	Fold51
62	Fold52
63	Fold53
64	Fold54
65	Fold55
66	Fold56
67	Fold57
68	Fold58
69	Fold59
70	Fold60
71	Fold61
72	Fold62
73	Fold63
74	Fold64
75	Fold65
76	Fold66
77	Fold67
78	Fold68
79	Fold69
80	Fold70
81	Fold71
82	Fold72
83	Fold73
84	Fold74
85	Fold75
86	Fold76
87	Fold77

88	Fold78
89	Fold79
90	Fold80
91	Fold81
92	Fold82
93	Fold83
94	Fold84
95	Fold85
96	Fold86
97	Fold87
98	Fold88
99	Fold89
100	Fold90
101	Fold91
102	Fold92
103	Fold93
104	Fold94
105	Fold95
106	Fold96
107	Fold97
108	Fold98
109	Fold99
110	Fold100

10. CancelBet Error Message

Error Message	Definition
PlacebetTimeout	Game provider did not receive PlaceBet/PlaceBetParlay/PlaceBet3rd response.
BettingFailed	System error.
OddsSuspend	Odds is temporarily closed.
MatchClosed	Match is closed.
OverMaxPerMatch	The stake is over than the limit setting of max per match.
OverBetLimit	The stake is over than the limit setting of max bet or less than the min bet.
OddsChanged	Odds is changed.
OverMaxPerBall	The stake is over than the limit setting of max per ball. Only applied on Number Game (SportType=161) and Happy5 (SportType=164).
PointChanged	The point of HDP/OU bet types is changed.
ClosePrice	Odds is temporarily closed.
OverRepeatedBet	The number of betting tickets is over than the limit setting in one bet type and match.
BetIntervalBlock	The betting interval seconds is less than the limit setting.
OverMaxPayoutPerMatch	The estimated payout is over than the limit setting of max payout per match.
MerchantResponseContentError	Merchant response incorrectly. E.g. Response not exist or null on refid when placebet.