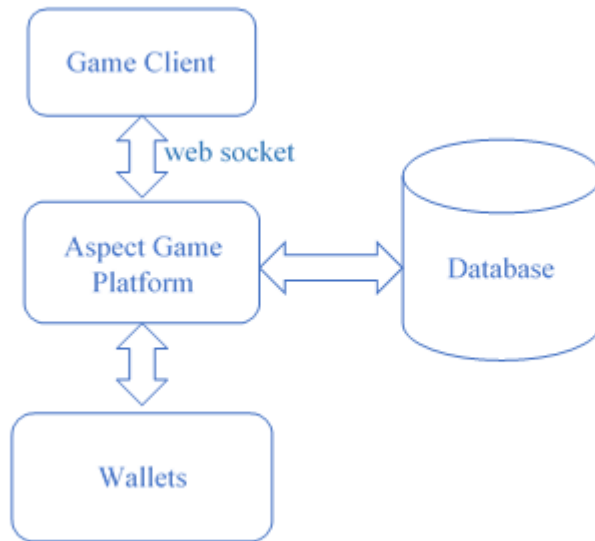


Aspect Game Platform Seamless Wallet Integration (v3.20.08)

- Aspect Game Platform Seamless Wallet Integration (v3.20.08)
- 1. Introduction
- 2. Wallet Call Workflow
 - 2.1 Loading game
 - 2.2 Every game round
- 3. Integration APIs
 - 3.0 Security
 - 3.1 URL Parameters
 - 3.2 authenticate
 - 3.3 balance
 - 3.4 debit
 - 3.5 credit
 - 3.6 rollback (Optional)
 - 3.7 completeToken (Optional)
- 4. Platform APIs
 - 4.1 Load game
 - 4.2 List games
 - 4.3 Show game history
 - 4.4 Get Games' RTP, stakes and max exposure
- 6. Wallet API Test
- Appendix
 - Currencies supported
 - Transfer Wallet Integration
 - Supported Game Languages
 - ErrorCode
 - Auto Complete Game Sessions

1. Introduction

All balances and amounts in this document are dollars.



2. Wallet Call Workflow

2.1 Loading game

- Wallet calls:
 - authenticate -> balance

2.2 Every game round

- Wallet calls:
 - debit -> credit
- Every game round must have and only have 1 debit and 1 credit call.
- Every game has a unique transactionKey and used by all wallet APIs.
- credit will always be sent for every game even win amount is 0.
- credit call also means game round is completed
- Example:
 - transactionKey in Debit: 6942310003415
 - transactionKey in Credit: 6942310003415

3. Integration APIs

3.0 Security

- All requests in production envs must have Authorization HTTP Header in following format:
 - **Authorization: AUTH {signature}**

- Authorization header is optional in UAT/Test/Dev envs. The {signature} must be correct if Authorization header is used.
- {signature} = md5({{secretKey}} + / + {api with parameters})
- Example:

```
1  {{secretKey}} = "test"
2  {{signature}} = md5("test/deposit?operatorId=1234&username=my.name&currenc
3      = "D70A675B0BB0610220730EE73063A85B"
4
5  HTTP header:
6  Authorization: AUTH D70A675B0BB0610220730EE73063A85B
```

3.1 URL Parameters

RequestParam	token	operatorId	gameId	amount	transactionKey	username
authenticate	✓	✓	✓			
debit	✓	✓	✓	✓	✓	
credit	✓	✓	✓	✓	✓	
balance	✓	✓	✓			
rollback	✓	✓	✓		✓	
completeToken		✓	✓		✓	✓

3.2 authenticate

- Request (HTTP Get):
 - **token**, **operatorId** and **gameId** are from loading URL params.

`https://${wallet}/authenticate?token=${token}&operatorId=${operatorId}&gameId=$`

- Response (Successful):

```
{
  "authenticated": true,
  "username": "uniqueName",
  "displayName": "nickName",
  "currency": "USD",
  "balance": 9605579.97,
  "playerGroup": "VIP"
}
```

- Response (Failed):

```
{
  "error": "Invalid Player",
  "errorCode": 1001
}
```

- Common Error Parameters

- error:
 - brief error message which can be any text.
- errorCode:
 - Will be sent to game client after converting (= -32400 - | errorCode % 100 |)

- Parameters

- authenticated
 - should always be true, otherwise should use Failed response format.
- username
 - Platform will use (operatorId + gameId + username) to load previous game state and record game history
- displayName (optional)
 - This value will be displayed in history page.
 - If displayName is not given, username will be used instead.
- currency (<https://docs.aspectgaming.dev/s/QxQXR6b3W>)
 - Follow ISO 4217 naming rule. The game client will use this currency to show meters. currency is case sensitive. usd and USD are 2 different currencies.
- balance
 - play's balance
- playerGroup (optional)
 - Every player group can have different stakes limit

3.3 balance

- Request (HTTP Get):

`https://{wallet}/balance?token=${token}&operatorId=${operatorId}&gameId=${gameId}`

- Response (Successful):

```
{
  "success": true,
  "balance": 9605579.97
}
```

- Response (Failed):

```
{
  "error": "Invalid Player",
  "errorCode": 1001
}
```

Note:

- Normally this API only need to be called one time when loading the game, but it may be called periodically according to operator's requirement.
- The balance will be updated by all the wallet calls.
- balance/amount are using dollars with 2 decimals.

3.4 debit

- Request (HTTP Post):

`https://{wallet}/debit?token=${token}&operatorId=${operatorId}&gameId=${gameId}`

1. With Headers:

```
"Content-Type": "application/x-www-form-urlencoded"
```

2. Parameters:

- token
- operatorId
- gameId
- amount : debit amount (unit: dollar; exp: 1.50 means 1.5 dollar)
- transactionKey

- Every wagered game has an unique transactionKey.

- Response (Successful):

```
{
  "success": true,
  "balance": 9605579.97
}
```

- Response (Failed):

```
{
  "error": "Invalid Player",
  "errorCode": 1001
}
```

3.5 credit

- Request (HTTP Post):

`https://${wallet}/credit?token=${token}&operatorId=${operatorId}&gameId=${gameI`



1. With Headers:

`"Content-Type": "application/x-www-form-urlencoded"`

2. Parameters:

- token
- operatorId
- gameId
- amount : credit amount (unit: dollar)
- transactionKey
 - If no response within 5 seconds, the same wallet call will be sent to wallet again.
 - Normally the credit call will be tried 2 times (configurable) until response received.

- Response (Successful):

```
{
  "success": true,
  "balance": 9605579.97
}
```

- Response (Failed):

```
{
  "error": "Invalid Player",
  "errorCode": 1001
}
```

3.6 rollback (Optional)

Rollback API will be invoked, if debit timeout or network error.

- Request (HTTP Post):

`https://${wallet}/rollback?token=${token}&operatorId=${operatorId}&gameId=${gam`



1. With Headers:

```
"Content-Type": "application/x-www-form-urlencoded"
```

2. Parameters

- token
- operatorId
- gameId
- transactionKey

- Response (Successful):

```
{
  "success": true,
  "balance": 9605579.97
}
```

- Response (Failed):

```
{
  "error": "Invalid Player",
  "errorCode": 1001
}
```

3.7 completeToken (Optional)

Our platform offers an auto-completion feature for games, ensuring that any unfinished rounds are automatically played and resolved within a specified time frame. For instance, should a player exit during a freeGame and remain absent for 24 hours, our system will intervene to finalize the game. This API is designed to retrieve a token necessary for completing any rounds left incomplete.

- Request (HTTP Post):

`https://${wallet}/completeToken?operatorId=${operatorId}&gameId=${gameId}&usern`



1. With Headers:

`"Content-Type": "application/x-www-form-urlencoded"`

2. Parameters:

- operatorId
- gameId
- username
- transactionKey

- Response (Successful):

```
{
  "success": true,
  "token": "xxx-xxx-xxx",
  "balance": 9605579.97
}
```

- Response (Failed):

```
{
  "error": "xxx",
  "errorCode": xxx
}
```


4. Platform APIs

4.1 Load game

- 3rd party integrator should load our game by using the following url
- Real Mode URL:

`https://${game-server}/agp-launcher/${gameId}/?token=${token}&operatorId=${oper`

- Fun Mode URL:

`https://${game-server}/agp-launcher/${gameId}/?mode=fun&language=${language}&ca`

game-server will be “uat.aspectgaming.com (http://uat.aspectgaming.com)” for Integration.

Example URL:

`https://uat.aspectgaming.com/agp-launcher/32/?mode=fun` (https://uat.aspectgaming.com/agp-launcher/32/?mode=fun)

- Parameters
 - token
 - This is a session token and it will be used by all wallet calls to identify the current session.
 - Wallet server should be able to identify the player, game, operator according to this token.
 - operatorId
 - Identify the operator, different operator may connect to different wallet server.
 - language
 - This is used to control the language in game UI. Currently, the following 6 languages are supported. Game will use English UI if the language parameter is not supported.
 - Chinese: **zh-CN**
 - English: **en-US**
 - Thai: **th-TH**
 - Indonesia: **id-ID**
 - Russian: **ru-RU**

- Turkish: **tr-TR**
- mode
 - mode=fun for fun mode, not need to fill token and operatorId
- Home Button Navigation Parameters
 - Parameters
 - **callbackUrl** or **lobbyUrl** (Optional): Defines the destination URL
 - **callbackType** (Optional): Specifies the navigation method
 - Display and Behavior Logic
 - When `callbackUrl` is provided:

The home button **will be displayed** with the following navigation behaviors:

callbackType	Navigation Behavior
tophref or not specified	Navigates top-level window: <code>window.top.location.href = callbackUrl</code>
parenthref	Navigates parent window: <code>window.parent.location.href = callbackUrl</code>
currenthref	Navigates current window: <code>window.location.href = callbackUrl</code>
navigate	Uses legacy navigation: <code>window.navigate(callbackUrl)</code>

- When `callbackUrl` is NOT provided:

The home button's behavior depends solely on `callbackType`:

callbackType	Navigation Behavior
topreload	Home button appears and refreshes top window: <code>window.top.location.reload()</code>
backreload	Home button appears, goes back, then refreshes: <code>window.history.back(); location.reload()</code>
not specified	No home button is displayed

4.2 List games

1. Usage

- Use this API to list all active games of current operator.
- Currently supported language: en-US and zh-CN.
- All languages start with zh will use zh-CN language.
- All unknown languages will use en-US language.

2. Request (Get):

```
1 GET {{game-server}}/agp-api/settings/games?operatorId={{operatorId}}&l
2 Content-Type: application/x-www-form-urlencoded
```

- operatorId: string
- language: string (optional, default: en-US)

3. Response

- games: array of json including following fields
 - gameId: string
 - gameName: string
 - type: String (Crash, Roulette, Keno, Arcade, Lottery, Slots)

4. Example

```
1 GET https://uat.aspectgaming.com/agp-api/settings/games?operatorId=8
2
3 {
4   "success":true,
5   "games":[
6     {"gameId":"1","gameName":"The Magic Brush","type":"Slots"},
7     {"gameId":"2","gameName":"Fortune Cats","type":"Slots"},
8     {"gameId":"3","gameName":"Backyard Gold","type":"Slots"},
9     ...
10  ]
11 }
12
13 GET https://uat.aspectgaming.com/agp-api/settings/games?operatorId=8&l
14
15 {
16   "success":true,
17   "games":[
18     {"gameId":"1","gameName":"妙笔生财","type":"Slots"},
19     {"gameId":"2","gameName":"喵喵进宝","type":"Slots"},
20     {"gameId":"3","gameName":"此地无银300两","type":"Slots"},
21     ...
22  ]
23 }
```

4.3 Show game history

1. Url:

```
1 | {{game-server}}/agp-history/history_detail.html?eventId={{transactionK
```



Params Name	Attribute	Value
eventId	Mandatory	transactionKey
language	Optional	en-US/zh-CN/th-TH/id-ID/ru-RU/tr-TR
timezoneOffset	Optional	number in minus

2. Example

```
1 | https://uat.aspectgaming.com/agp-history/history_detail.html?eventId=1
```



Test Game Detail

Game Name : Jin Qian Bao

Round ID : 1902967496231619930

Currency : GBP

Total Bet : £1.20

Number Of Paylines : 30

Player : eric.chang

Bet Multiplier : £0.04

Total Win : £8.80

Game ID : 34

Game Result

Time (+08:00)	Conduct	Amount (GBP)
2022-08-23T10:45:47.600+08:00	Bet	£1.20
2022-08-23T10:45:47.606+08:00	Win	£8.80

Base Game

Click to show win lines



Total Bet : £1.20

Win : £8.80

Bet Time : 2022-08-23T10:45:47.600+08:00

Win Line Number : 3



PayLine	Length	Symbol	Multiplier	Win
8	3		1	£0.40
15	3		1	£0.40
20	5		1	£8.00

4.4 Get Games' RTP, stakes and max exposure

1. Usage

- Use this API to list all active games's RTP, stakes and max exposure.
-

supported language: en-US and zh-CN.

- All languages start with zh will use zh-CN language.
- All unknown languages will use en-US language.

2. Request (Get):

```
1 GET {{game-server}}/agp-api/settings/stakes?operatorId={{operatorId}}&  
2 Content-Type: application/x-www-form-urlencoded
```

- operatorId: string (mandatory)
- currency: string (mandatory)
- language: string (optional, default: en-US)

3. Response

- games: array of json including following fields
 - operatorId: string
 - gameId: string
 - gameName: string
 - rtp: string
 - maxWinX: Number
 - stakeConfig: JsonObject
 - currency
 - stakes: string
 - defaultStake: Number
 - min: Number
 - max: Number
 - maxWinAmount: Number

4. Example

```
1 GET https://uat.aspectgaming.com/agp-api/settings/stakes?operatorId=8&
2
3 [
4   {
5     "operatorId": "8",
6     "gameId": "1",
7     "gameName": "The Magic Brush",
8     "rtp": "94.24",
9     "maxWinX": 852,
10    "stakeConfig": {
11      "USD": {
12        "stakes": "0.5,1.0,2.0,4.0,6.0,10.0,25.0,50.0,75.0,125.0",
13        "defaultStake": 1.0,
14        "min": 0.5,
15        "max": 125.0,
16        "maxWinAmount": 106500.0
17      }
18    }
19  },
20  ...
```

5. Download csv

Add format=csv at the end of request string.

```
1 GET https://uat.aspectgaming.com/agp-api/settings/stakes?operatorId=8&
```

6. Wallet API Test

Please help to configure your **endpoint**, **operatorId** and **token** before API test.

Settings:

wallet endpoint

operatorId

gameId

debitAmount

creditAmount

transactionKey

token

secretKey

CORS Proxy

Value:

1

1.01

1.02

1746626718458000001

Off

Method:

authenticate

balance

debit

credit

rollback

completeToken

Test:

execute Type1 (GET)

execute (GET)

execute (POST)

execute (POST)

execute (POST)

execute (POST)

Result:

N/A

N/A

N/A

N/A

N/A

N/A

statusCode:

Balance:

Request:

Response:

Log:

Appendix

Currencies supported

click to see more all the currencis including cryptos we support:
<https://docs.aspectgaming.dev/s/QxQXR6b3W> (<https://docs.aspectgaming.dev/s/QxQXR6b3W>)

Transfer Wallet Integration

click to see more details:
<https://docs.aspectgaming.dev/s/JKrVfmw4P> (<https://docs.aspectgaming.dev/s/JKrVfmw4P>)

Supported Game Languages

Current support:

- Chinese: **zh-CN**
- English: **en-US**
- Thai: **th-TH**
- Indonesia: **id-ID**
- Russian: **ru-RU**
- Turkish: **tr-TR**

ErrorCode

Please respond with network status 200 if you want us to report the error and stop the current round.

If you have an internal server error and you want us to retry the same request, please respond with a different status code than 200.

We will retry any request except debit up to 2 times (total 3 times) for the same session.

If the wallet request fails 3 times, the game client will be disconnected.

The most retry times and interval time can also be configured

Error Code	Error	Applicable Methods
1001	Invalid Player	authenticate
1002	Invalid Token	All
1008	Invalid Game ID	authenticate
1012	Game Unavailable	authenticate
1003	Insufficient Funds	debit
1028	Duplicate Transaction Key	debit
1026	Amount cannot be negative	debit, credit
1023	No Debit Before Credit	credit
1029	Transaction Not Found	credit
1030	Round has been canceled	credit
1024	Could Not Rollback After Credit	Rollback
1025	No Debit Transaction To Rollback	Rollback

- It is suggested that the wallet should add more detailed error codes to identify different errors.
- The wallet should add the error reason in the response message body as “error”.

```
{  
  "error": "Invalid Player",  
  "errorCode": 1001  
}
```

- Platform Error Codes

In order to standarize the handling of important errors at game client side, we will map some error codes from wallet to our Platform Error Codes

(<https://docs.aspectgaming.dev/s/r6QQ40qBL>)

Auto Complete Game Sessions

Some operators might require the game providers to complete the game sessions immediately after the session is disconnected by any reasons.

For more details about auto complete feature, please refer to:

<https://docs.aspectgaming.dev/s/S1EMjzu74> (<https://docs.aspectgaming.dev/s/S1EMjzu74>)