



## API2ALL Integration Document

### HISTORY CHANGES DOCUMENTS

\*A - **Add new**; M - **Modify**; D - **Delete**

Day change	Location change	A* M D	Source	Old version	Describe change	New version
2020-12-30	Transfer Wallet Integration	A	V2.5	V2.5	<ul style="list-style-type: none"><li>• Search API</li><li>• SearchDetail API</li><li>• Update API</li></ul>	<b>V2.6</b>
2020-12-14	Player Integration	M	V2.4	V2.4	Register/Create player API: return the redirect URL	<b>V2.5</b>
2020-12-07	Player Integration	A	V2.3	V2.3	New get outstanding ticket API	<b>V2.4</b>
2020-12-01	Game provider	A	V2.2	V2.2	New vendors: * MT: support <b>THB currency only</b> * Habanero	<b>V2.3</b>

2020-11-02	Player Integration	M	V2.1	V2.1	Open game API: return the game URL for * Pragmatic ( <b>MX</b> ) * Pragmatic Play Live ( <b>PPL</b> )	<b>V2.2</b>
2020-10-19	* Player Integration * API Endpoint	A	V2.0	V2.0	#1. New API: Get member list by creation time #2. API endpoints	<b>V2.1</b>
2020-10-13	* Game provider * Player Integration	A	V1.9	V1.9	#1. Add new vendor: PGSOFT e-game #2. Login API: required player's IP info	<b>V2.0</b>
2020-09-28	Player Integration	A	V1.8	V1.8	New APIs: #1. Vendor's maintenance status #2. Fetch ticket by time and paging	<b>V1.9</b>
2020-09-23	Game provider	A	V1.7	V1.7	Add new vendor: QQ lottery	<b>V1.8</b>
2020-09-18	Player Integration	A	V1.6	V1.6	New APIs: #1. System maintenance status #2. Kick out players	<b>V1.7</b>
2020-08-21	Transfer wallet integration	A	V1.5	V1.5	New API: get Deposit/Withdrawal history by time	<b>V1.6</b>

2020-08-14	Player Integration	M	V1.5	V1.5	#1. <a href="#">Fetch ticket by time</a> : update limit rate 3s each time to send the request #2. Get ticket API : no more functioning	<b>V1.5</b>
2020-08-07	*Player Integration *Game provider	A	V1.4	V1.4	New API: <a href="#">Fetch ticket by time</a> New vendor: Pragmatic Live Casino	<b>V1.5</b>
2020-07-16	Player Integration	A	V1.3	V1.3	Add new API: <a href="#">Get Outstanding Ticket</a>	<b>V1.4</b>
2020-07-07	Game provider	M	V1.3	V1.3	Remove unavailable vendor: <ul style="list-style-type: none"> <li>• LX (Lixin)</li> <li>• MS (HongKong6)</li> <li>• SSC (SSC)</li> </ul>	
2020-06-29	*Game provider *Open Game	M	V1.2	V1.2	Update Ezugi vendor Update <a href="#">product</a> for each vendor Update Result and Referer on Open Game API	<b>V1.3</b>
2020-06-22	Game provider	D	N/A	V1.1	Remove unavailable vendor <ul style="list-style-type: none"> <li>• XPG</li> <li>• BETSOFT</li> </ul>	<b>V1.2</b>

## HISTORY CHANGES DOCUMENTS

### Player Integration

1. Register/Create Player
2. Login
3. Open Game
4. Get Balance
5. Get Game List
6. Get WinLose
7. Change password
8. Get Outstanding
9. Fetch ticket by time
10. System maintenance status
11. Kick out players
12. Vendor's maintenance status
13. Fetch paging
14. Get member list by creation time

### Transfer Wallet Integration

1. Deposit
2. Withdrawal
3. Check transaction status
4. Deposit/Withdrawal history
5. Search ticket
6. Search ticket details
7. Update ticket status

## [Appendix](#)

[Authenticate algorithm](#)

[Create Sign](#)

[Language code](#)

[Game provider](#)

[API2ALL Products](#)

[Statement date](#)

[Error code](#)

[API Endpoint](#)

# Player Integration

## 1. Register/Create Player

- Request : **{Protocol}://{Endpoint}/api/partner/register**
  - Protocol : **HTTP**
  - Method : **POST**
  - Content-Type : **application/json**
  - Endpoint : **Authentication** endpoint, provided by API2ALL company
- This method using to register new player in API2ALL's database system
- Post data model

Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at partner side when send request
UserName	string	Login name of player. Username must be between 5 and 15 characters. Contain include only letters (a-z) or numbers (0-9).
Password	string	Password of player Between 8 - 15 characters Combination of alphabetic: <b>characters</b> and <b>numbers</b> Valid characters: <b>a-z A-Z 0-9 !@#-_.?</b>
Fullname	string	Full name of player
Email	string	Valid email of player
Mobile	string	Mobile phone of player
Gender	integer	<ul style="list-style-type: none"> <li>• <b>-1</b> : unknow</li> <li>• <b>0</b> : female</li> <li>• <b>1</b> : male</li> </ul>

DoB	string	Date of birthday in format <b>yyyy-MM-dd</b>
Currency	string	Valid currency code
BankName	string	Valid bank name
BankAccountNo	string	Serial bank account number

- Response data format

Name	Data Type	Description
Error	integer	<b>Error code.</b> Negative means fail operation. Zero means successfully operation.
Message	string	Error message in text format
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at API2ALL's side when send response

- Example: using Postman to call API2ALL's API

api2.data333.com/api + ... **Request** No Environment ⌵ 👁 ⚙

**POST** ⌵ api2.data333.com/api/partner/register Params **Send** ⌵ Save ⌵

Authorization Headers (1) **Body** Pre-request Script Tests Code

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ⌵

```
1 {
2   "Partner": "",
3   "Sign": "70f82393425b8a2c0b42b21906a6ff23e5f8d8be2cf3c26a9edf6e089f1576dc",
4   "TimeStamp": 1586191392,
5   "UserName": "001ply",
6   "Password": "qwer1234",
7   "Fullname": "Tester 01",
8   "Email": "@gmail.com",
9   "Mobile": "0123456789",
10  "Gender": 0,
11  "DoB": "1983-03-03",
12  "Currency": "CNY",
13  "BankName": "CH",
14  "BankAccountNo": "5687413354858"
15 }
16
```

**Response** Status: 200 OK Time: 4988 ms

Body Cookies (1) Headers (13) Test Results

Pretty Raw Preview JSON ⌵ 🔍

```
1 {
2   "Token": null,
3   "Error": 0,
4   "Message": "Success Operation",
5   "Sign": "70f82393425b8a2c0b42b21906a6ff23e5f8d8be2cf3c26a9edf6e089f1576dc",
6   "TimeStamp": 1586191392,
7   "UTC": "2020-04-06T09:43:41.2895279Z"
8 }
```

## 2. Login

- Request : **{Protocol}://{Endpoint}/api/partner/login**
  - Protocol : **HTTP**



- Method : **POST**
- Content-Type : **application/json**
- Endpoint : **Authentication** endpoint, provided by API2ALL company
- This method using for player to login into API2ALL system
- Note: partner no need to send the plain text password to API2ALL's API
- Post data model

Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at partner side when send request
UserName	string	<ul style="list-style-type: none"> <li>● Login name of player.</li> <li>● Start with alphabet charater, contains only character and digit.</li> </ul>
IP	string	Player's IP

- Response data format

Name	Data Type	Description
------	-----------	-------------

Error	integer	<b>Error code.</b> Negative means fail operation. Zero means successfully operation.
Message	string	Error message in text format
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at API2ALL's side when send response
Token	string	Session token key of player in API2ALL's system


- Example: using Postman to call API2ALL's API


POST <http://pauthapi.linkv2.com/api/partner/login>

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON**

```
1 {
2   "UserName": " ",
3   "Partner": " ",
4   "Sign": "802f8f28bbe1f02601a1cd54a7b4c56e156186048a44097d186cbc738e8328f9",
5   "TimeStamp": 1606060881,
6   "IP": "9.8.7.6"
7 }
```

Body Cookies (1) Headers (15) Test Results  Status: 200 OK

Pretty Raw Preview Visualize **JSON** 

```
1 {
2   "Token": "f8ab4306-d49b-4f91-9ab0-8d2c73dd7bca",
3   "RedirectUrl": " // /login?token=f8ab4306-d49b-4f91-9ab0-8d2c73dd7bca&lang=EN",
4   "Error": 0,
5   "Message": "Success Operation",
6   "Sign": "802f8f28bbe1f02601a1cd54a7b4c56e156186048a44097d186cbc738e8328f9",
7   "TimeStamp": 1606060881,
8   "UTC": "2020-12-14T04:34:55.3730027Z"
9 }
```

### 3. Open Game

- Request: **{Protocol}://{Endpoint}/api/play/login**
  - Protocol : **HTTP**
  - Method : **POST**
  - Content-Type : **application/json**

○ Endpoint : **Open game** endpoint, provided by API2ALL company

○ Headers :

■ **Authorization : Bearer {token}**

■ **Referer**

- This method using for player to open game
- Post data model

Name	Data Type	Description
Vendor	string	<b>Game provider</b> > <b>VendorCode</b> , will be provided by API2ALL company
Device	string	optional
Browser	string	Browser that used by player to open game
GameCode	string	Game that player want to open and play
GameHall	string	Using for <b>CQ9</b> game only, default value is “CQ9”
Lang	string	<b>Language code</b>
MerchantCode	string	Using to open <b>Skywind</b> slot game only
Ticket	string	Using to open <b>Pragmatic</b> game only
SeatId	string	Using to open <b>Lucky Streak</b> live casino game only
Tag	string	Optional

GameProvider	string	Optional
--------------	--------	----------

- Response data format

Name	Data Type	Description
Error	integer	<b>Error code.</b> Negative means fail operation. Zero means successfully operation.
Message	string	Error message in text format
Success	boolean	Action result
Result	object	<ul style="list-style-type: none"> <li>• Data: Game URL returned</li> <li>• Settings: player setting <ul style="list-style-type: none"> <li>◦ MaxPerBet: bet limit on game (apply for Live Casino product)</li> </ul> </li> </ul>
TargetUrl	string	URL extend information

- Example: using Postman to call API2ALL's API

POST `opengameapi.data333.com/api/play/login`

Params Authorization Headers (12) Body Pre-request Script Tests Settings

Headers `10 hidden`

KEY	VALUE
<input checked="" type="checkbox"/> Authorization	Bearer 9d70a9-822c-4a71-887f-d66dd3bbe993
<input checked="" type="checkbox"/> Referer	http://119.81.200.187/sport

Open game

POST `opengameapi.data333.com/api/play/login`

Params Authorization Headers (12) Body Pre-request Script Tests Settings

Headers `10 hidden`

KEY	VALUE
<input checked="" type="checkbox"/> Authorization	Bearer 9d70a9-822c-4a71-887f-d66dd3bbe993
<input checked="" type="checkbox"/> Referer	http://119.81.200.187/sport

Body

```
1 {
2   "Vendor": "YT",
3   "Device": "unknown",
4   "Browser": "chrome",
5   "GameCode": "36",
6   "GameHall": "",
7   "Lang": "en-us",
8   "MerchantCode": "",
9   "Ticket": "",
10  "SeatId": "",
11  "Tag": "",
12  "GameProvider": ""
13 }
```

Body Cookies (1) Headers (16) Test Results

Status: 200 OK Time: 1569 ms Size: 790 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "Result": {
3     "Data": "http://test-client.virtual-technology.com/36/?version=1.0&language=en&channelId=56718&data=173657y8u10gCY0K2fsvstQKVLc1f1a1242K30hCupCAK&D16n90mfS3U2K2bo",
4     "Settings": {
5       "MaxPerBet": -1.0
6     }
7   },
8   "TargetUrl": null,
9   "Success": true,
10  "Error": null,
11  "Message": null
12 }
```

POST `opengameapi.data333.com/api/play/login`

Params Authorization Headers (12) Body Pre-request Script Tests Settings

Headers `10 hidden`

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> Authorization	Bearer 0701035a-d4aa-48a6-8375-1665b1c86d10	
<input checked="" type="checkbox"/> Referer	http://119.81.200.187/sport	
Key	Value	Description

Body Cookies (1) Headers (15) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "Result": "https://w4cd007d.mywinday.com/public/auth.aspx?us=a81a1ntmyr002p1y&k=0ABC3C77B8EC4AA99F0D1C54E2ECFF1A&lang=ZH-CN&accType=MY&r=443499141&ref=",
3   "TargetUrl": null,
4   "Success": true,
5   "Error": null,
6   "Message": null
7 }
```

POST `opengameapi.data333.com/api/play/login`

Params Authorization Headers (12) Body Pre-request Script Tests Settings

Headers `10 hidden`

KEY	VALUE
<input checked="" type="checkbox"/> Authorization	Bearer 0701035a-d4aa-48a6-8375-1665b1c86d10
<input checked="" type="checkbox"/> Referer	http://119.81.200.187/sport

Body

```
1 {
2   "Vendor": "M8",
3   "Device": "unknown",
4   "Browser": "chrome",
5   "GameCode": "",
6   "GameHall": "",
7   "Lang": "zh-cn",
8   "MerchantCode": "",
9   "Ticket": "",
10  "SeatId": "",
11  "Tag": "",
12  "GameProvider": ""
13 }
```

## 4. Get Balance

- Request : **{Protocol}://{Endpoint}/api/partner/balance**
  - Protocol : **HTTP**
  - Method : **POST**
  - Content-Type : **application/json**
  - Endpoint : **Authentication** endpoint, provided by API2ALL company
- Using to get the current balance of player.
- Post data model

Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner
Sign	string	A hashed string that will be generated by using <a href="#">CreateSign</a> method
TimeStamp	long	Unix time at partner's side when send request
UserName	string	User name of player

- Response data format

Name	Data Type	Description
------	-----------	-------------

Error	integer	<b>Error code.</b> Negative means fail operation. Zero means successfully operation.
Message	string	Error message in text format
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at API2ALL's side when send response
Balance	decimal(18,6)	Balance of player

- Example: using Postman to call API2ALL's API



The screenshot displays a REST client interface with two main sections: 'Request' and 'Response'.

**Request Section:**

- Method:** POST
- URL:** api2.data333.com/api/partner/balance
- Body Type:** JSON (application/json)
- Body Content:**

```
1 {  
2   "Partner": "  
3   "Sign": "d60c9bfbba1996887025f0a87ef111eb346c7df06d452956f995c4fa8f43e504",  
4   "TimeStamp": 1586462954,  
5   "Username": "  
6 }  
7
```

**Response Section:**

- Body Type:** JSON
- Body Content:**

```
1 {  
2   "Balance": 35,  
3   "Error": 0,  
4   "Message": "Success Operation",  
5   "Sign": "d60c9bfbba1996887025f0a87ef111eb346c7df06d452956f995c4fa8f43e504",  
6   "TimeStamp": 1586462954,  
7   "UTC": "2020-04-09T13:09:26.9780516Z"  
8 }
```

## 5. Get Game List

- Request : **{Protocol}://{Endpoint}/api/gamelist/find**
  - Protocol : **HTTP**
  - Method : **GET**

- Headers : **Authorization : Bearer {token}**
- Endpoint : **Game list** endpoint, provided by API2ALL company
- Using to get the game list provided.
- Query String Parameters

Name	Data Type	Description
Vendor	string	<b>Game provider</b> > <b>VendorCode</b> , will be provided by API2ALL company

- Response data format

Name	Data Type	Description
ErrorCode	integer	<b>Error code.</b> Negative means fail operation. Zero means successfully operation.
ErrorMessage	string	Error message in text format
Data	string	Game information detail
TargetUrl	string	Url extend
Success	boolean	Action result
Message	string	Extend message info

- Example: using Postman to call API2ALL's API

GET
gamelistsapi.data333.com/api/gamelists/find?Vendor=DPT
Request

Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings

Headers
8 hidden

KEY	VALUE
<input checked="" type="checkbox"/> Authorization	Bearer 0701035a-d4aa-48a6-8375-1665b1c86d10
Key	Value

Body
Cookies (1)
Headers (14)
Test Results
Response

Pretty
Raw
Preview
Visualize
JSON

```

1 {
2   "Result": {
3     "ErrorMessage": "OK",
4     "ErrorCode": 0,
5     "Data": [
6       {
7         "gameID": "pt_hb",
8         "gameName": "A Night Out",
9         "gameTypeID": "vs",
10        "typeDescription": "Video Slot",
11        "technology": "html5",
12        "technologyID": "H5",
13        "platform": "MOBILE,WEB",
14        "demoGameAvailable": true,
15        "gameIdNumeric": 85
16      },
17      {
18        "gameID": "pt_aeolus",
19      },
20      {
21        "gameID": "sw_111",
22        "gameName": "Long Long Long",
23        "gameTypeID": "vs",
24        "typeDescription": "Video Slot",
25        "technology": "html5",
26        "technologyID": "H5",
27        "platform": "MOBILE,WEB",
28        "demoGameAvailable": false,
29        "gameIdNumeric": 12
30      },
31      {
32        "gameID": "pt_pnn",
33        "gameName": "Panther Moon",
34        "gameTypeID": "vs",
35        "typeDescription": "Video Slot",
36        "technology": "html5",
37        "technologyID": "H5",
38        "platform": "MOBILE,WEB",

```

## 6. Get WinLose

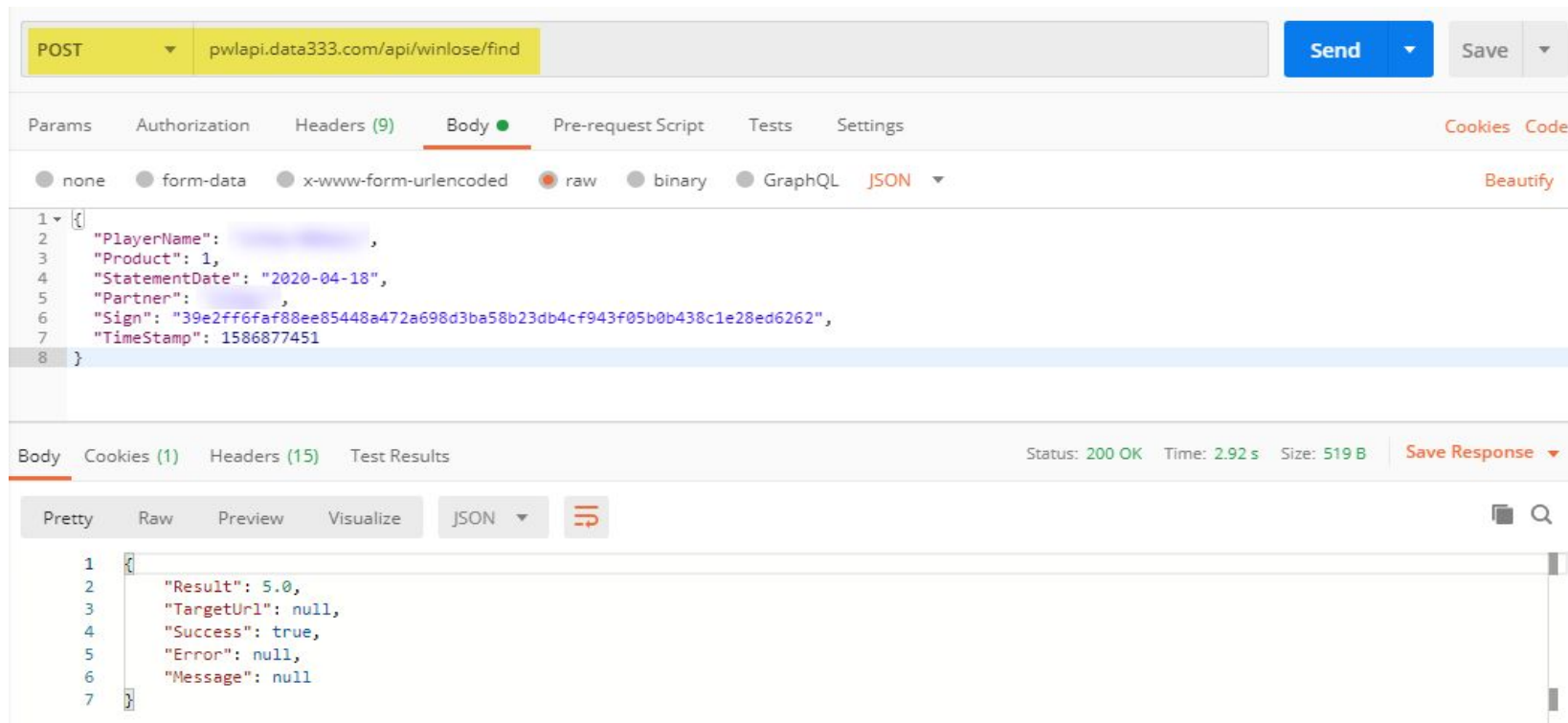
- Request : **{Protocol}://{Endpoint}/api/winlose/find**
  - Protocol : **HTTP**
  - Method : **POST**
  - Content-Type : **application/json**
  - Endpoint : **WinLose/Ticket** endpoint, provided by API2ALL company
- Using to get player win/lose on API2ALL game product with specific statement date
- Post data model

Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at partner's side when send request
Product	integer	<b>API2ALL game product</b> , provided by API2ALL company
PlayerName	string	Username of player
StatementDate	string	API2ALL <b>statement date</b>

- Response data format

Name	Data Type	Description
Error	integer	<b>Error code.</b> Negative means fail operation. Zero means successfully operation.
Message	string	Error message in text format
TargetUrl	string	Url extend
Success	boolean	Action result
Result	string	List of ticket info

- Example: using Postman to call API2ALL's API



## 7. Change password

- Request
  - Protocol : **HTTP**
  - Method : **POST**
  - Content-Type : **application/json**
  - Endpoint : **Authentication** endpoint, provided by API2ALL company

- Using to change the password for specific player
- Post data model

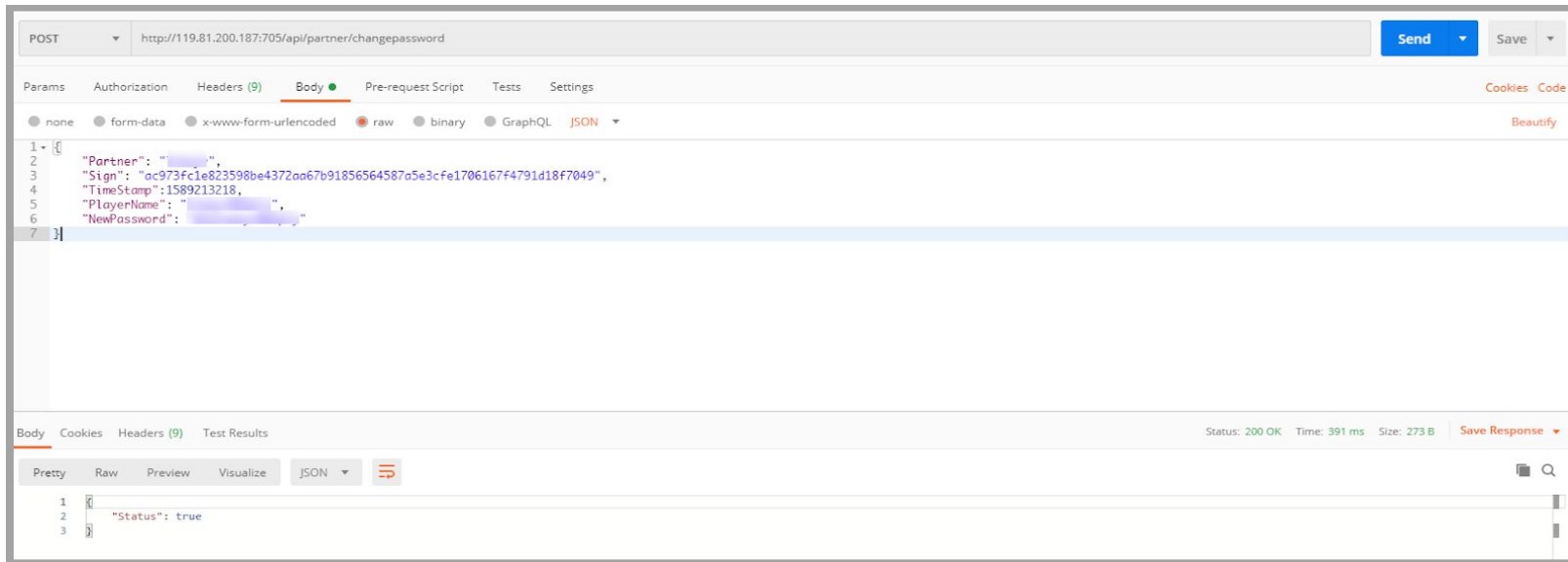
Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at partner's side when send request
PlayerName	string	Username of player
NewPassword	string	<ul style="list-style-type: none"> <li>• Player's new password</li> <li>• Required</li> <li>• Max length: 15</li> <li>• Min length: 8</li> </ul>

- Response data format

Name	Data Type	Description
Status	boolean	<ul style="list-style-type: none"> <li>• <b>TRUE</b> : action success</li> <li>• <b>FALSE</b> : action fail</li> </ul>



- Example: using Postman to call API2ALL's API



## 8. Get Outstanding

- Request : **{Protocol}://{Endpoint}/api/tickets/findOutstandingTickets**
  - Protocol : **HTTP**
  - Method : **POST**
  - Content-Type : **application/json**
  - Endpoint : **WinLose/Ticket** endpoint, provided by API2ALL company
- Using to get Outstanding ticket of all players on all games
- Post data model

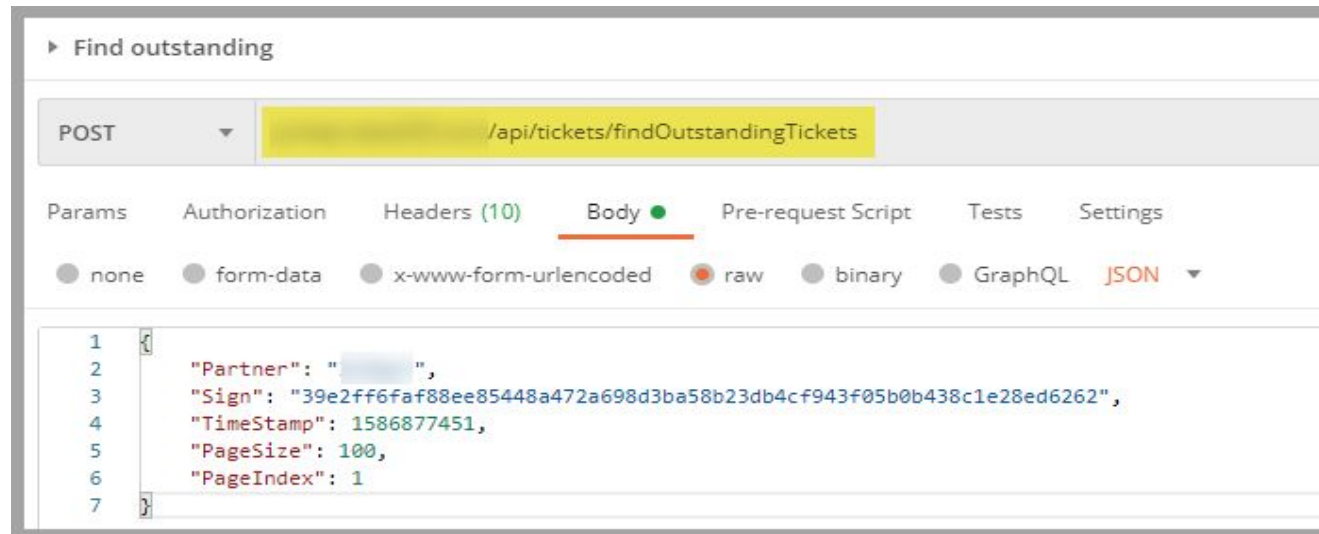
Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at partner's side when send request
PageSize	int	Record number that partner want to get
PageIndex	int	Page number

- Response data format

Name	Data Type	Description
Error	object	<b>Error code.</b> Negative means fail operation. Zero means successfully operation.
Message	string	Error message in text format
TargetUrl	string	Url extend
Success	boolean	Action result

Result	object	List of ticket info
--------	--------	---------------------

- Example: using Postman to call API2ALL's API



Result

Tickets

0

- ID : 94397
- Vendor : 1
- Product : 0
- TicketSessionID : "0"
- RawTicketID : 0
- TicketID : "M8\_HDP4309334345"
- MemberName : "Intmyr011ply"
- VendorPlayerName : "Intmyr011ply"
- GameType : "HDP"
- RoundID : "04152SV82949WI"
- StakeMYR : 0
- Stake : 5
- StakeMoney : 5
- StakeMoneyMYR : 0
- WinlossMYR : 0
- WinLoss : 0
- Result : ""
- Currency : "MYR"
- VendorTicketDateUTC : "2020-06-04T07:29:50"
- VendorTicketDate : "2020-06-04T15:29:50"
- StatementDate : "2020-06-04T00:00:00"
- PlayerID : 104673
- PlayerWinLoss : 0
- PlayerWinLossMYR : 0
- PlayerCommission : 0
- PlayerCommissionAmount : 0
- PlayerCommissionAmountMYR : 0
- AgentID : 0
- AgentWinLoss : 0
- AgentWinLossMYR : 0
- AgentCommission : 0
- AgentCommissionAmount : 0
- AgentCommissionAmountMYR : 0
- MasterAgentID : 0
- MasterAgentWinLoss : 0
- MasterAgentWinLossMYR : 0
- MasterAgentCommission : 0
- MasterAgentCommissionAmount : 0
- MasterAgentCommissionAmountMYR : 0
- SeniorID : 0
- SeniorWinLoss : 0
- SeniorWinLossMYR : 0
- SeniorCommission : 0
- SeniorCommissionAmount : 0
- SeniorCommissionAmountMYR : 0
- SuperSeniorID : 104521
- SuperSeniorWinLoss : 0
- SuperSeniorWinLossMYR : 0
- SuperSeniorCommission : 0
- SuperSeniorCommissionAmount : 0
- SuperSeniorCommissionAmountMYR : 0
- AdminID : 63722
- AdminWinLoss : 0
- AdminWinLossMYR : 0
- AdminCommission : 0
- AdminCommissionAmount : 0
- AdminCommissionAmountMYR : 0
- CompanyID : 3
- CompanyWinLoss : 0
- CompanyWinLossMYR : 0
- CompanyCommission : 0
- CompanyCommissionAmount : 0

## 9. Fetch ticket by time

- Request : **{Protocol}://{Endpoint}/api/tickets/fetch**
  - Protocol : **HTTP**
  - Method : **POST**
  - Content-Type : **application/json**
  - Endpoint : **WinLose/Ticket** endpoint, provided by API2ALL company
- Desc:
  - Using to get the list processed ticket of all player on all game providers with a time period.
  - Ticket return have
    - ProcessedTime **>=** StartTime
    - ProcessedTime **<** EndTime
    - ProcessedTime : the time that system processed ticket done
  - The max time range (time between StartTime and EndTime) is **30 seconds**.
  - The time interval per request is **3 seconds** (e.g: the 1st time to send the request is **12:00:00**, need wait at least 3s to send next request, the 2nd time can do send the request is **12:00:04**)
- Post data model

Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner

Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at partner's side when send request
StartTime	string	* API2ALL system processed time * Format: <b>yyyy-MM-dd HH:mm:ss</b> * Time zone: <b>UTC</b>
EndTime	string	* API2ALL system processed time * Format: <b>yyyy-MM-dd HH:mm:ss</b> * Time zone: <b>UTC</b>

- Response data format

Name	Data Type	Description
Error	Object	<b>Error code.</b> Negative means fail operation. Zero means successfully operation.
Message	string	Error message in text format
TargetUrl	string	Url extend
Success	boolean	Action result
Result	Object	List of ticket info

- Example: using Postman to call API2ALL's API

Fetch ticket by time

POST /api/tickets/fetch

Params Authorization Headers (10) Body Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "Partner": " ",
3   "sign": "e8c86e71ebd3773c969ed8f97c677298089125560c6267969c95a38948f57535",
4   "TimeStamp": 1596645523,
5   "StartTime": "2020-08-06 04:23:00",
6   "EndTime": "2020-08-06 04:23:30"
7 }
```

```

{
  "Result": {
    "Tickets": [
      {
        "Vendor": 1,
        "Product": 0,
        "TicketSessionID": "sample string 1",
        "TicketID": "sample string 2",
        "MemberName": "sample string 3",
        "GameType": "sample string 4",
        "RoundID": "sample string 5",
        "Stake": 6.0,
        "StakeMoney": 7.0,
        "Result": "sample string 8",
        "Currency": "sample string 9",
        "VendorTicketDateUTC": "2020-08-07T18:59:13.3699914+08:00",
        "VendorTicketDate": "2020-08-07T18:59:13.3699914+08:00",
        "StatementDate": "2020-08-07T18:59:13.3699914+08:00",
        "PlayerWinLoss": 13.0,
        "PlayerCommissionAmount": 14.0,
        "ProcessedTime": "2020-08-07T18:59:13.3699914+08:00",
        "CreatedTime": "2020-08-07T18:59:13.3699914+08:00",
        "Ip": "sample string 17",
        "ValidAmount": 18.0,
        "Type": 19
      },
      {
        "Vendor": 1,
        "Product": 0,
        "TicketSessionID": "sample string 1",
        "TicketID": "sample string 2",
        "MemberName": "sample string 3",
        "GameType": "sample string 4",
        "RoundID": "sample string 5",
        "Stake": 6.0,
        "StakeMoney": 7.0,
        "Result": "sample string 8",
        "Currency": "sample string 9",
        "VendorTicketDateUTC": "2020-08-07T18:59:13.3699914+08:00",
        "VendorTicketDate": "2020-08-07T18:59:13.3699914+08:00",
        "StatementDate": "2020-08-07T18:59:13.3699914+08:00",
        "PlayerWinLoss": 13.0,
        "PlayerCommissionAmount": 14.0,
        "ProcessedTime": "2020-08-07T18:59:13.3699914+08:00",
        "CreatedTime": "2020-08-07T18:59:13.3699914+08:00",
        "Ip": "sample string 17",
        "ValidAmount": 18.0,
        "Type": 19
      }
    ]
  }
}

```



## 10. System maintenance status

- Request : **{Protocol}://{Endpoint}/api/maintenance/status**
  - Protocol : **HTTP**
  - Method : **POST**
  - Content-Type : **application/json**
  - Endpoint : **Authentication** endpoint, provided by API2ALL company
- Desc: using to check that API2ALL system under maintenance or not
- Post data model

Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at partner's side when send request

- Response data format

Name	Data Type	Description
Status	int	<ul style="list-style-type: none"><li>• <b>0</b> : system alive</li></ul>

		<ul style="list-style-type: none"> <li>• <b>1</b>: system under maintenance</li> </ul>
Error	Object	<b>Error code.</b> Negative means fail operation. Zero means successfully operation.
Message	string	Error message in text format

- Example: using Postman to call API2ALL's API

► Maintenance status


POST /api/maintenance/status

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼

```
1 {
2   "Partner": ,
3   "sign": "a3ba63f55dce3485513985efa9bdd50572cc5cbc7d40e93f959c820439111ff7",
4   "TimeStamp": 1600278335
5 }
```

Body Cookies (1) Headers (15) Test Results

Pretty Raw Preview Visualize **JSON** 

```
1 {
2   "status": 0,
3   "Error": 0,
4   "Message": "Success Operation",
5   "sign": null,
6   "TimeStamp": 0,
7   "UTC": "2020-09-18T09:21:12.5874523Z"
8 }
```

## 11. Kick out players

- Request : **{Protocol}://{Endpoint}/api/partner/kick**
  - Protocol : **HTTP**
  - Method : **POST**
  - Content-Type : **application/json**
  - Endpoint : **Authentication** endpoint, provided by API2ALL company
- Desc: using to kick out all logged in players
- Post data model

Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at partner's side when send request

- Response data format

Name	Data Type	Description
Error	int	<b>Error code.</b>

		Negative means fail operation. Zero means successfully operation.
Message	string	Error message in text format

- Example: using Postman to call API2ALL's API

► Kick out players


POST /api/partner/kick

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼

```
1 {  
2   "Partner": ,  
3   "Sign": "a3ba63f55dce3485513985efa9bdd50572cc5cbc7d40e93f959c820439111ff7",  
4   "Timestamp": 1600278335  
5 }
```

Body Cookies (1) Headers (15) Test Results

Pretty Raw Preview Visualize JSON ▼ 

```
1 {  
2   "Error": 0,  
3   "Message": "Success Operation",  
4   "Sign": null,  
5   "Timestamp": 0,  
6   "UTC": "2020-09-18T09:31:45.5974463Z"  
7 }
```

## 12. Vendor's maintenance status

- Request : **{Protocol}://{Endpoint}/api/maintenance/vendorstatus**
  - Protocol : **HTTP**
  - Method : **POST**
  - Content-Type : **application/json**
  - Endpoint : **Authentication** endpoint, provided by API2ALL company
- Desc: checking the maintenance status for a certain vendor
- Post data model

Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at partner's side when send request
Vendor	string	<b>Game provider</b> > <b>VendorCode</b> , will be provided by API2ALL company

- Response data format

Name	Data Type	Description
Status	int	The maintenance status code <ul style="list-style-type: none"> <li>Active: <b>0</b></li> <li>Maintenance: <b>-1</b></li> </ul>
Error	int	<b>Error code.</b> Negative means fail operation. Zero means successfully operation. <ul style="list-style-type: none"> <li>Success Operation : <b>0</b></li> <li>Invalid Parameter : <b>-2</b></li> <li>Authentication Failed : <b>-3</b></li> <li>Invalid Vendor : <b>-4</b></li> </ul> <p><i>*please ignore Status code when response with Error Code not 0 means fail to check status</i></p>
Message	string	Error message in text format
UTC	string	Processed request time on UTC

- Example: using Postman to call API2ALL's API



Vendor maintenance status

POST /api/maintenance/vendorstatus

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded **raw** binary GraphQL JSON

```
1 {
2   "Partner": ,
3   "Sign": "a3ba63f55dce3485513985efa9bdd50572cc5cbc7d40e93f959c820439111ff7",
4   "TimeStamp": 1600278335,
5   "Vendor": "MS"
6 }
```

Body Cookies (1) Headers (16) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "Error": 0,
3   "Message": "Success Operation",
4   "Status": 0,
5   "UTC": "2020-09-28T09:59:58.1864528Z"
6 }
```

## 13. Fetch paging

- Request : **{Protocol}://{Endpoint}/api/tickets/fetchpaging**
  - Protocol : **HTTP**
  - Method : **POST**
  - Content-Type : **application/json**
  - Endpoint : **WinLose/Ticket** endpoint, provided by API2ALL company
- Desc:
  - Using to get the list processed ticket of all player on all game providers
    - On a time period.
    - Have page size
  - Ticket return have
    - ProcessedTime **>=** StartTime
    - ProcessedTime **<** EndTime
    - ProcessedTime : the time that system processed ticket done
  - The time interval per request is **3 seconds** (e.g: the 1st time to send the request is **12:00:00**, need wait at least 3s to send next request, the 2nd time can do send the request is **12:00:04**)
- Post data model

Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL

		company to partner
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at partner's side when send request
FromDate	string	* API2ALL system processed time * Format: <b>yyyy-MM-dd HH:mm:ss</b> * Time zone: <b>UTC</b>
ToDate	string	* API2ALL system processed time * Format: <b>yyyy-MM-dd HH:mm:ss</b> * Time zone: <b>UTC</b>
MetaData	object	* An object that help system know the last offset to get corresponding ticket * Value get from response
PageSize	int	* The ticket number that API2ALL returns * Maximum number: <b>500</b>

- Response data format

Name	Data Type	Description
Error	Object	<b>Error code.</b> Negative means fail operation. Zero means successfully operation.

Message	string	Error message in text format
TargetUrl	string	Url extend
Success	boolean	Action result
Result	Object	* Tickets: List of ticket info * Metadata: An object that help system know the last offset to get corresponding ticket. Passing on request to pull next tickets

- Example: using Postman to call API2ALL's API

Fetch ticket by time + paging

### Request

POST /api/tickets/fetchpaging

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "FromDate": "2020-09-28 10:45:00",
3   "ToDate": "2020-09-28 10:59:00",
4   "Metadata": {
5     "FirstSortOffset": [
6       {}
7     ],
8     "LastSortOffset": [
9       {}
10    ],
11    "PageIndex": 0,
12    "Direction": 0
13  },
14  "PageSize": 3,
15  "Partner": "",
16  "Sign": "15b329082403f349a52135364c93d7faf04f1c97c6b1f8de266662cf7d70a032",
17  "TimeStamp": 1601315751
18 }
```

no passing Metadata

### Response

Result

Tickets

0

- Vendor : 45
- Product : 4
- TicketSessionID : "4652-8276-29ae841df115"
- TicketID : "795347"
- MemberName : ""
- GameType : "pt\_bfb"
- RoundID : "424795347"
- Stake : 0.4
- StakeMoney : 0.4
- Result : ""
- Currency : "MYR"
- VendorTicketDateUTC : "2020-09-28T10:50:20.625Z"
- VendorTicketDate : "2020-09-28T18:50:20.625+08:00"
- StatementDate : "2020-09-28T00:00:00"
- PlayerWinLoss : -0.4
- PlayerCommissionAmount : 0
- ProcessedTime : "2020-09-28T18:50:21.189337+08:00"
- CreatedTime : "2020-09-28T18:50:20.625+08:00"
- Ip : ""

Fetch ticket by time + paging

**Request**

POST /api/tickets/fetchpaging

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1  {
2    "FromDate": "",
3    "ToDate": "",
4    "Metadata": {
5      "FirstSortOffset": [
6        1601290090137,
7        1601290088941,
8        "dpt_424791008"
9      ],
10     "LastSortOffset": [
11       1601290100157,
12       1601290090702,
13       "dpt_424791311"
14     ],
15     "PageIndex": 0,
16     "Direction": 0
17   },
18   "PageSize": 3,
19   "Partner": "",
20   "Sign": "15b329082483f349a52135364c93d7faf04f1c97c6b1f8de266662cf7d70a032",
21   "TimeStamp": 1601315751
22 }

```

passing Metadata

**Response**

Result

Tickets

0

- Vendor : 45
- Product : 4
- TicketSessionID : "4208-8a18-4f3e18c11783"
- TicketID : "4791509"
- MemberName :
- GameType : "pt\_bfb"
- RoundID : "424791509"
- Stake : 0.4
- StakeMoney : 0.4
- Result : ""
- Currency : "MYR"
- VendorTicketDateUTC : "2020-09-28T10:48:24.831Z"
- VendorTicketDate : "2020-09-28T18:48:24.831+08:00"
- StatementDate : "2020-09-28T00:00:00"
- PlayerWinLoss : -0.2
- PlayerCommissionAmount : 0

## 14. Get member list by creation time

- Request : **{Protocol}://{Endpoint}/api/partner/getmemberlist**
  - Protocol : **HTTP**
  - Method : **POST**
  - Content-Type : **application/json**
  - Endpoint : **Authentication** endpoint, provided by API2ALL company
- Desc: get member list info that created on a time period (from time, to time)
- Post data model

Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at partner's side when send request
FromUtcDate	string	* The date that player created on * Format: <b>yyyy-MM-dd</b> * Time zone: <b>UTC</b>
ToUtcDate	string	* The date that player created on * Format: <b>yyyy-MM-dd</b> * Time zone: <b>UTC</b>

PageSize	int	Record number that partner want to get
PageIndex	int	Page number

- Response data format

Name	Data Type	Description
Error	int	* <b>Error code.</b> * Negative means fail operation. * Zero means successfully operation.
Message	string	Error message in text format
Members	object	List of member's info

- Example: using Postman to call API2ALL's API



Get member list by creation time

POST `/api/partner/getmemberlist`

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON**

```
1 {
2   "Partner": " ",
3   "Sign": "9aae0249f526fbf413e7edadac422fc9ad1af206831fea4b2689c026053cc166",
4   "TimeStamp": 1603106962,
5   "FromUtcDate": "2020-10-08",
6   "ToUtcDate": "2020-10-09",
7   "PageSize": 100,
8   "PageIndex": 1
9 }
10
```

**RESPONSE**

```
1 {
2   "Error": 0,
3   "Message": "Success Operation",
4   "Sign": "9aae0249f526fbf413e7edadac422fc9ad1af206831fea4b2689c026053cc166",
5   "TimeStamp": 1603106962,
6   "Members": [
7     {
8       "Name": "test ",
9       "Currency": "MYR",
10      "BankName": "HONG LEONG",
11      "BankAccount": "123",
12      "BankBranch": "kl",
13      "Email": "",
14      "Mobile": "",
15      "Extra1": "",
16      "Extra2": "",
17      "DOB": "2020-10-08T00:00:00",
18      "Remark": "",
19      "Status": 0,
20      "CreationDateUTC": "2020-10-08T07:57:47"
21    }
22  ]
23 }
```

## 15. Get outstanding by player

- Request : **{Protocol}://{Endpoint}/api/tickets/findPlayerTickets**
  - Protocol : **HTTP**
  - Method : **POST**
  - Content-Type : **application/json**
  - Endpoint : **WinLose/Ticket** endpoint, provided by API2ALL company
- Using to get Outstanding ticket of a specific player.
- Post data model

Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at partner's side when send request
Vendor	int	<ul style="list-style-type: none"><li>• Vendor ID (refer <b>ID</b> value on <b>Game provider</b>)</li><li>• <b>Optional</b></li></ul>
PlayerName	string	Username of player

- Response data format

Name	Data Type	Description
Error	object	<b>Error code.</b> Negative means fail operation. Zero means successfully operation.
Message	string	Error message in text format
TargetUrl	string	Url extend
Success	boolean	Action result
Result	object	List of ticket info

- Example: using Postman to call API2ALL's API

Get outstanding by player

NO passing Vendor param

POST /api/tickets/findPlayerTickets

Send Save

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Beautify

```
1 {
2   "Partner": ,
3   "TimeStamp": 1607357113,
4   "Sign": "eee430c6119422f3ec0abad7550434b3f68c02887f7595d1e5e72de70dd6277d",
5   "PlayerName": 
6 }
```

Body Cookies (1) Headers (15) Test Results

Status: 200 OK Time: 120 ms Size: 6.33 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "Result": {
3     "Tickets": [
4       {
5         "ID": 17922922,
6         "Vendor": 1,
7         "Product": 1,
8         "TicketSessionID": "e227c868-e24e-4d33-8013-b96ebba96cf8",
9         "RawTicketID": 0,
10        "TicketID": "MS_OU4604918294",
11        "MemberName": ,
12        "VendorPlayerName": ,
13        "GameType": "OU",
14        "RoundID": "07170TG60455DA",
15        "StakeMYR": 0.0,
16        "Stake": 5.000000,
17        "StakeMoneyMYR": 5.000000,
18        "StakeMoneyMYR": 0.0,
19        "WinLossMYR": 0.0,
20        "WinLoss": 0.000000,
21        "Result": "",
22        "Currency": "MYR",
23        "VendorTicketDateUTC": "2020-12-07T09:04:55",
24        "VendorTicketDate": "2020-12-07T17:04:55"
```

Get outstanding by player

POST `apl/tickets/findPlayerTickets` Send Save

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON**

Body

```
1 {
2   "Partner": " ",
3   "TimeStamp": "1607357113",
4   "sign": "eee430c6119422f3ec0abad7550434b3f68c02887f7595d1e5e72de70dd8277d",
5   "Vendor": 2,
6   "PlayerName": " "
7 }
```

Body Cookies (1) Headers (15) Test Results

Status: 200 OK Time: 86 ms Size: 3.43 KB Save Response

Pretty Raw Preview Visualize **JSON**

```
1 {
2   "Result": {
3     "Tickets": [
4       {
5         "ID": 17922697,
6         "Vendor": 2,
7         "Product": 1,
8         "TicketSessionID": "21d1f07c-6117-44b0-820f-d1ed38acd436",
9         "RawTicketID": 0,
10        "TicketID": "ssport_0010808662",
11        "MemberName": " ",
12        "VendorPlayerName": " ",
13        "GameType": "OU",
14        "RoundID": "",
15        "StakeMYR": 0.0,
16        "Stake": 3.000000,
17        "StakeMoney": 3.000000,
18        "StakeMoneyMYR": 0.0,
19        "WinlossMYR": 0.0,
20        "Winloss": 3.000000,
21        "Result": "",
22        "Currency": "MYR",
23        "VendorTicketDateUTC": "2020-12-07T08:59:53",
24        "VendorTicketDate": "2020-12-07T16:59:53"
```

# Transfer Wallet Integration

## 1. Deposit

- Request : **{Protocol}://{Endpoint}/rest/transfer/deposit**
  - Protocol : **HTTP**
  - Method : **POST**
  - Content-Type : **application/json**
  - Endpoint : **Transfer wallet** endpoint, provided by API2ALL company
- This method using to deposit the credit to player's wallet.
- Post data model

Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner
Sign	string	A hashed string that will be generated by using <a href="#">CreateSign</a> method
TimeStamp	long	Unix time at partner side when send request
TransactionId	string	Unique transaction id from partner side

Player	string	Username of player
Amount	decimal(18,6)	Deposit amount, is <b>positive number</b>

- Response data format

Name	Data Type	Description
Error	integer	<b>Error code.</b> Negative means fail operation. Zero means successfully operation.
Message	string	Error message in text format
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at API2ALL's side when send response
Player	string	Username of player
Amount	decimal(18,6)	Deposit amount, is <b>positive number</b>
Before	decimal(18,6)	Balance before deposit
After	decimal(18,6)	Balance after deposit

- Example: using Postman to call API2ALL's API

POST w2.data333.com/rest/transfer/deposit Request Params Send Save

Authorization Headers (1) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```

1 {
2   "Partner": "",
3   "Sign": "7a6e6a7deb7550dc5882d8032140b26030edd331506d30a3c316051c9ec2eb99",
4   "TimeStamp": 1586446661,
5   "Player": "002ply",
6   "TransactionId": "",
7   "Amount": 50
8 }
9

```

Body Cookies (1) Headers (14) Test Results Response Status: 200 OK Time: 1511 ms

Pretty Raw Preview JSON

```

1 {
2   "Player": "002ply",
3   "Amount": 50,
4   "Before": 0,
5   "After": 50,
6   "Error": 0,
7   "Message": "Success Operation",
8   "Sign": null,
9   "TimeStamp": 0,
10  "UTC": "2020-04-09T08:44:33.951061Z"
11 }

```

## 2. Withdrawal

- Request : {Protocol}://{Endpoint}/rest/transfer/withdrawal
  - Protocol : HTTP



- Method : **POST**
- Content-Type : **application/json**
- Endpoint : **Transfer wallet** endpoint, provided by API2ALL company
- This method using to withdraw the credit from player's wallet.
- Post data model

Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at partner side when send request
TransactionId	string	Unique transaction id from partner side
Player	string	Username of player
Amount	decimal(18,6)	Withdrawal amount, is <b>positive number</b>

- Response data format

Name	Data Type	Description
Error	integer	<b>Error code.</b>

		Negative means fail operation. Zero means successfully operation.
Message	string	Error message in text format
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at API2ALL's side when send response
Player	string	Username of player
Amount	decimal(18,6)	Deposit amount, is <b>positive number</b>
Before	decimal(18,6)	Balance before deposit
After	decimal(18,6)	Balance after deposit

- Example: using Postman to call API2ALL's API

POST w2.data333.com/rest/transfer/withdrawal Request Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "Partner": " ",
3   "Sign": "7a6e6a7deb7550dc5882d8032140b26030edd331506d30a3c316051c9ec2eb99",
4   "TimeStamp": 1586446661,
5   "Player": " ",
6   "TransactionId": " ",
7   "Amount": 15
8 }
9
```

Body Cookies (1) Headers (14) Test Results Response Status: 200 OK Time: 1031 ms

Pretty Raw Preview JSON

```
1 {
2   "Player": " ",
3   "Amount": 15,
4   "Before": 50,
5   "After": 35,
6   "Error": 0,
7   "Message": "Success Operation",
8   "Sign": null,
9   "TimeStamp": 0,
10  "UTC": "2020-04-09T08:55:31.7160499Z"
11 }
```

### 3. Check transaction status

- Request : {Protocol}://{Endpoint}/rest/transfer/check
  - Protocol : HTTP

- Method : **POST**
- Content-Type : **application/json**
- Endpoint : **Transfer wallet** endpoint, provided by API2ALL company
- This method using to check status of a certain transaction.
- Post data model

Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at partner side when send request
TransactionId	string	Unique transaction id from partner side

- Response data format

Name	Data Type	Description
Error	integer	<b>Error code.</b> Negative means fail operation. Zero means successfully operation.
Message	string	Error message in text format

Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at API2ALL's side when send response
Status	integer	<ul style="list-style-type: none"> <li>• <b>0</b> : fail</li> <li>• <b>1</b> : success</li> <li>• <b>-1</b> : not existed</li> </ul>

#### 4. Deposit/Withdrawal history

- Request : **{Protocol}://{Endpoint}/rest/transfer/transferlogs**
  - Protocol : **HTTP**
  - Method : **POST**
  - Content-Type : **application/json**
  - Endpoint : **Transfer wallet** endpoint, provided by API2ALL company
- This method using to get deposit/withdrawal history of player with a time period.
- Post data model

Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method

TimeStamp	long	Unix time at partner side when send request
Player	string	Username of player
FromDate	string	format <b>yyyy-MM-dd</b>
ToDate	string	format <b>yyyy-MM-dd</b>

- Response data format

Name	Data Type	Description
Error	integer	<b>Error code.</b> Negative means fail operation. Zero means successfully operation.
Message	string	Error message in text format
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at API2ALL's side when send response
Result	object	List of transactions
UTC	string	Processed request done on UTC time

- Example: using Postman to call API2ALL's API

Deposit/Withdraw history

POST /rest/transfer/transferlogs

Params Authorization Headers (10) Body Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "Partner": " ",
3   "Player": " ",
4   "FromDate": "2020-08-11",
5   "ToDate": "2020-08-11",
6   "TimeStamp": 1598025982,
7   "Sign": "f09690e3906634b1d75c380a9d438e5ca5029d2f3d0b6498b8e6e02104a5429b"
8 }
```

```

1  {
2    "Result": [
3      {
4        "PartnerName": " ",
5        "Sign": "5d68fb17f37b7c3370a19fac2ad6a155ff5c68f4df26684902c4b0403d4bfe7b",
6        "TimeStamp": 1597115083,
7        "TransactionId": "2020081110044343",
8        "Player": " ",
9        "Amount": 100.000000,
10       "BeforeBalance": 0.154200,
11       "AfterBalance": 100.154200,
12       "CreatedDateUTC": "2020-08-11T03:04:43",
13       "ErrorCode": 0
14     },
15     {
16       "PartnerName": " ",
17       "Sign": "5f76c143af2386ec0a3b531cbc50f0eb8cff620a7c19b6dc059f112f8a4b16de",
18       "TimeStamp": 1597145929,
19       "TransactionId": "2020081106384949",
20       "Player": " ",
21       "Amount": 200.000000,
22       "BeforeBalance": 6.904200,
23       "AfterBalance": 206.904200,
24       "CreatedDateUTC": "2020-08-11T11:38:49",
25       "ErrorCode": 0
26     },
27     {
28       "PartnerName": " ",
29       "Sign": "717d2da5657292fdfb180ff1405f6f00409627fd617ca5ef75b01f2af10ae4e6",
30       "TimeStamp": 1597147694,
31       "TransactionId": "2020081107081414",
32       "Player": " ",
33       "Amount": 200.000000,
34       "BeforeBalance": 102.151700,
35       "AfterBalance": 302.151700,
36       "CreatedDateUTC": "2020-08-11T12:08:14",
37       "ErrorCode": 0
38     },
39     {
40       "PartnerName": " ",
41       "Sign": "fc83b3568b951a69b75cc561f8c78d1ab2bc4165c3e4bf205ef9ba5d409955ae",
42       "TimeStamp": 1597147699,
43       "TransactionId": "2020081107081919",
44       "Player": " ",
45       "Amount": 302.000000,
46       "BeforeBalance": 302.151700,
47       "AfterBalance": 0.151700,
48       "CreatedDateUTC": "2020-08-11T12:08:19",
49       "ErrorCode": 0
50     }
51   ],
52   "Error": 0,
53   "Message": null,
54   "Sign": "f09690e3906634b1d75c380a9d438e5ca5029d2f3d0b6498b8e6e02104a5429b",
55   "TimeStamp": 1598025982.

```



## 5. Search ticket

- Request : **{Protocol}://{Endpoint}/rest/cash-ticket/search**
  - Protocol : **HTTP**
  - Method : **POST**
  - Content-Type : **application/json**
  - Endpoint : **Transfer wallet** endpoint, provided by API2ALL company
- This method using to list out player's deposit/withdrawal tickets.
- Post data model

Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at partner side when send request
Date	string	Statement date
Type	int	<b>1</b> =Deposit; <b>2</b> =Withdraw
Status	int	<b>0</b> =Pending; <b>2</b> =Approved; <b>-1</b> =Rejected

- Response data format

Name	Data Type	Description
Error	integer	<b>Error code.</b> Negative means fail operation. Zero means successfully operation.
Message	string	Error message in text format
Records	object	List of transactions

- Example: using Postman to call API2ALL's API

POST http://[redacted]rest/cash-ticket/search Send Save

Params Authorization Headers (11) **Body** Pre-request Script Tests Settings Cookies Code

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☒ JSON Beautify

```

1 {
2   "date": "2020-12-29",
3   "type": 1,
4   "Status": 0,
5   "partner": "[redacted]",
6   "sign": "28757d297c2114b84d2b0f5c4020883b17340a02609df064a430f58dc01f8f4c",
7   "timestamp": 123
8 }

```

**Body** Cookies (1) Headers (17) Test Results Status: 200 OK Time: 298 ms Size: 1.12 KB Save Response

Pretty Raw Preview Visualize JSON ≡

```

1 {
2   "Result": {
3     "Records": [
4       {
5         "Id": 104637,
6         "TransactionNo": "4UICKD04C9XC6",
7         "PlayerId": 136264,
8         "Type": 2,
9         "TransferMethod": 0,
10        "PayGate": 0,
11        "Bank": "MyBank",
12        "Currency": "MYR",
13        "Amount": 12.000000,
14        "Status": 1,
15        "CreatedDate": "2020-12-29T18:45:49",
16        "CreatedDateUtc": "2020-12-29T10:45:49Z",
17        "BankBranch": "",
18        "BankAccountNo": "[redacted]",
19        "BankAccountName": "[redacted]"

```

## 6. Search ticket detail

- Request : **{Protocol}://{Endpoint}/rest/cash-ticket/searchdetail**

- Protocol : **HTTP**
- Method : **POST**
- Content-Type : **application/json**
- Endpoint : **Transfer wallet** endpoint, provided by API2ALL company
- This method using to list out player's deposit/withdrawal tickets with detail information.
- Post data model

Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at partner side when send request
Date	string	Statement date
Type	int	<b>1</b> =Deposit; <b>2</b> =Withdraw
Status	int	<b>0</b> =Pending; <b>2</b> =Approved; <b>-1</b> =Rejected

- Response data format

Name	Data Type	Description
------	-----------	-------------

Error	integer	<b>Error code.</b> Negative means fail operation. Zero means successfully operation.
Message	string	Error message in text format
Records	object	List of transactions

- Example: using Postman to call API2ALL's API

POST http://[redacted]/rest/cash-ticket/searchdetail Send Save

Params Authorization Headers (11) **Body** Pre-request Script Tests Settings Cookies Code

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```

1 {
2   "date": "2020-12-29",
3   "type": 1,
4   "status": 0,
5   "partner": "[redacted]",
6   "sign": "28757d297c2114b84d2b0f5c4020883017340e02609df064a430f58dc01f8f4c",
7   "timestamp": 123
8 }

```

**Body** Cookies (1) Headers (17) Test Results Status: 200 OK Time: 207 ms Size: 1.61 KB Save Response

Pretty Raw Preview Visualize JSON ≡ 🔍

```

1 {
2   "Result": {
3     "Records": [
4       {
5         "Id": 104637,
6         "PlayerId": 136264,
7         "TransferType": 2,
8         "TransferMethod": 0,
9         "Status": 1,
10        "CreatedBy": "[redacted]",
11        "ApprovedBy": "",
12        "ApprovedDate": null,
13        "ApprovedDateUtc": null,
14        "RejectedBy": "",
15        "RejectedDate": null,
16        "RejectedDateUtc": null,
17        "Amount": 12.000000,
18        "Currency": "MYR",
19        "CreatedDate": "2020-12-29T18:45:49",
20        "CreatedDateUtc": "2020-12-29T18:45:49Z",
21        "TransferDate": "2020-12-29T18:45:49",
22        "TransferDateUtc": "2020-12-29T18:45:49",
23        "PlayerTransferDate": "",
24        "TransactionNo": "4UCK004C9XCG",
25        "PayGate": 0,
26        "PayGateStatus": "",
27        "PayGateBankCode": "",
28        "BankName": "MyBank",
29        "BankAccountName": "[redacted]",
30        "BankAccountNo": "44444333",
31        "Remarks": "test api update withdrawal ticket",
32        "CurrentRollover": 0.0,
33        "TargetRollover": 0.0,
34        "ReceiptNo": "",
35        "PlayerBalance": 140.000000,
36        "BankBranch": "",
37        "PackageType": 0,
38        "VendorAmount": 0.0,
39        "VendorExchangeRate": 0.0,
40        "VendorPayGateType": 0
41      }
42    ]
43  },
44  "TargetUrl": null,
45  "Success": true,

```

## 7. Update ticket status

- Request : **{Protocol}://{Endpoint}/rest/cash-ticket/update**
  - Protocol : **HTTP**
  - Method : **POST**
  - Content-Type : **application/json**
  - Endpoint : **Transfer wallet** endpoint, provided by API2ALL company
- This method using to approve/reject a deposit/withdrawal ticket.
- Post data model

Name	Data Type	Description
Partner	string	Partner's name that will be provided by API2ALL company to partner
Sign	string	A hashed string that will be generated by using <b>CreateSign</b> method
TimeStamp	long	Unix time at partner side when send request
TicketId	long	Ticket Id
Remarks	string	Remark
Status	int	<b>0</b> =Pending; <b>2</b> =Approved; <b>-1</b> =Rejected
IP	string	IP address

- Response data format

Name	Data Type	Description
Error	integer	<b>Error code.</b> Negative means fail operation. Zero means successfully operation.
Message	string	Error message in text format
Success	boolean	Action result

- Example: using Postman to call API2ALL's API



POST http://[redacted]/rest/cash-ticket/update Send Save

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings Cookies Code

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "ticketid": 104648,
3   "remarks": "test",
4   "status": 2,
5   "ip": "1.2.3.4",
6   "partner": "[redacted]",
7   "sign": "28757d297c2114b84d2b0f5c4020883017340e02609df064a430f58dc01f8f4c",
8   "timestamp": 123
9 }
```

Body Cookies (1) Headers (17) Test Results Status: 200 OK Time: 1569 ms Size: 821 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "Result": 0,
3   "TargetUrl": null,
4   "Success": true,
5   "Error": {
6     "Code": 0,
7     "Message": "success"
8   },
9   "Message": null
10 }
```

# Appendix

## Authenticate algorithm

- API2ALL system will provide to partner data as example below
  - Partner name : **aaa**
  - Partner key : **FB67717C-26AE-4B01-BCA2-09DBC1A19A50**
  - Hash algorithm : **SHA256**
- Every API call, partner need to create a hashed string (sign) using
  - TimeStamp : Unix time stamp in partner system
  - Partner name : provided by API2ALL company
  - Partner key : provided by API2ALL company
- Create sign string by using [method](#) as below

## Create Sign

- A method using to generate a hashed string with hash algorithm **SHA256**

- Must have parameters : **Time Stamp, Partner Info, Partner Key**
- C# sample:

0 references | 0 changes | 0 authors, 0 changes

```
public static string CreateSign(this string timeStamp, string info, string key)
{
    var word = info.ToLower() + timeStamp + key.ToLower();

    using (var crypt = new SHA256Managed())
    {
        var hash = new StringBuilder();
        var crypto = crypt.ComputeHash(Encoding.UTF8.GetBytes(word), 0, Encoding.UTF8.GetByteCount(word));

        foreach (var theByte in crypto)
        {
            hash.Append(theByte.ToString("x2"));
        }

        return hash.ToString();
    }
}
```

- Using for: Register API, Deposit API, Withdrawal API, Get Ticket API, Get WinLose API, Get Outstanding API, Fetch Ticket API, Deposit/Withdrawal history, System maintenance status, Kick out players, Vendor's maintenance status, Fetch paging, Get member list by creation time, Get outstanding by player
  - Input: Time Stamp, Partner Name, Partner Key

- C# sample :

```
var pn = PARTNER_NAME;  
var sign = SignFunctions.CreateSign(TIME_STAMP, pn, PARTNER_KEY);
```

- Using for: [Login API](#)

- Input: Time Stamp, Partner Name, Partner Key, Player Name, Player Password
- C# sample :

```
var pn = PARTNER_NAME + PLAYER_NAME + PLAYER_PASSWORD;  
var sign = SignFunctions.CreateSign(TIME_STAMP, pn, PARTNER_KEY);
```

- Using for: [Get Balance](#)

- Input: Time Stamp, Partner Name, Partner Key, Player Name
- C# sample :

```
var pn = PARTNER_NAME + PLAYER_NAME;  
var sign = SignFunctions.CreateSign(TIME_STAMP, pn, PARTNER_KEY);
```

- Using for: [Change password API](#)

- Input: Time Stamp, Partner Name, Partner Key, Player Name, Player New Password

- C# sample :

```
var pn = PARTNER_NAME + PLAYER_NAME + PLAYER_NEW_PASSWORD;  
var sign = SignFunctions.CreateSign(TIME_STAMP, pn, PARTNER_KEY);
```

## Language code

- ISO language code
- Support
  - en-us : English
  - zh-cn : 简体中文
  - th-th : ภาษาไทย
  - ko-kr : 한국어
  - vi-vn : Tiếng Việt
  - id-id : Indonesia
  - hi-in : इंडिया

- **Game provider**

ID	VendorCode	VendorName	Product
----	------------	------------	---------

41	IA	IA E-Sports	Sport
1	M8	M-Sports	Sport
51	MaxBet	Max Bet	Sport
52	SBO	SBO	Sport
54	SBOVS	SBOVS	Sport
2	SSport	S-Sports	Sport
60	MT	Muay Thai	Sport
5	AB	AllBet	LiveCasino
7	AG	AG Deluxe Suite	LiveCasino
46	BG	BG Live Casino	LiveCasino
17	CF	CockFight	LiveCasino
35	CT855	CT855	LiveCasino
29	DG	Dream Gamming	LiveCasino
43	EVO	Evolution Live Casino	LiveCasino
4	EZUGI	EZUGI	LiveCasino
40	FGG	Fair Guaranteed Gaming	LiveCasino
31	GD	Gold Diamond Gaming	LiveCasino
23	LS	Lucky Streak	LiveCasino
56	PPL	Pragmatic Play Live	LiveCasino
36	SA	SA Gaming	LiveCasino
32	SB	Sexy Baccarat	LiveCasino
22	WM	WM	LiveCasino
61	MIKI	Miki Mouse	LiveCasino
48	C93	93 Connect	Lottery

25	KN	Keno Lottery	Lottery
26	ML	Malaysia Lottery (ABS)	Lottery
57	QQ	QQLottery	Lottery
16	ACE333	ACE333	Games
47	BGE	BG E-Game	Games
49	BS	BetSoft	Games
20	CQ9	CQ9	Games
45	DPT	PlayTech Digital	Games
50	DT	DreamTech	Games
21	GG	Global Gaming	Games
53	IG	IG E-Game	Games
44	JOKER	JOKER	Games
15	MX	Pragmatic	Games
9	PT	SkyWind	Games
39	SAE	SA Gaming EGames	Games
33	VT	Virtual Tech	Games
58	PGSOFT	PGSOFT	Games
59	Habanero	Habanero	Games
38	LEG	LE Gaming	P2P
37	OG	OG Poker	P2P

- **API2ALL Products**

Name	Value
Sport	1
LiveCasino	2
Lottery	3
Games	4
P2P	6
Financial	7

- **Statement date**

- Counted from **12:00:0 GMT+8** to **11:59:59 GMT+8**
- Example: statement date **2020-01-01** count from **2020-01-01 12:00:00** to **2020-01-02 11:59:59**

- **Error code**

Code	Message
-1	Invalid partner
-2	Invalid Sign
-3	Invalid player
-4	Partner account suspended
-5	Player suspended
-101	Player name invalid



-102	Password invalid
-104	Invalid company
-105	Not allow register
-106	Bank info is empty
-109	Invalid operation
-110	Email invalid
-111	Mobile invalid
-116	Invalid transaction
-1000	Invalid currency
-2002	Invalid partner
-2003	Not a cash admin
-2005	Invalid player name format
-2007	Password length invalid
-2008	Bad password
-2009	Player name existed
-2010	Currency not belong to company
-2011	Invalid time-zone
-2012	Invalid vip
-2013	Invalid operation
-2020	Invalid operation
-2021	Invalid operation
-2022	Invalid operation
-2023	Invalid operation
-2888	Invalid affiliate code

## ● API Endpoint

### ○ Production Environment

<b>1. Authentication</b>	<a href="http://pauthapi.linkv2.com">http://pauthapi.linkv2.com</a>
<b>2. Transfer wallet</b>	<a href="http://pwalletapi.linkv2.com">http://pwalletapi.linkv2.com</a>
<b>3. Winlose/Ticket</b>	<a href="http://pwlapi.linkv2.com">http://pwlapi.linkv2.com</a>
<b>4. Game list</b>	<a href="http://gamelistapi.linkv2.com">http://gamelistapi.linkv2.com</a>
<b>5. Open game</b>	<a href="http://opengameapi.linkv2.com">http://opengameapi.linkv2.com</a>

### ○ Staging Environment

<b>1. Authentication</b>	<a href="http://pauthapi.data333.com">http://pauthapi.data333.com</a>
<b>2. Transfer wallet</b>	<a href="http://w2.data333.com">http://w2.data333.com</a>
<b>3. Winlose/Ticket</b>	<a href="http://pwlapi.data333.com">http://pwlapi.data333.com</a>
<b>4. Game list</b>	<a href="http://gamelistapi.data333.com">http://gamelistapi.data333.com</a>
<b>5. Open game</b>	<a href="http://opengameapi.data333.com">http://opengameapi.data333.com</a>

6. Player	<a href="http://api2.data333.com">http://api2.data333.com</a>
-----------	---