Contents

1. Document History 2
2. API Overview 3
3. API Login 4
   3.1. Access Token Login 4
   3.2. Normal Login 5
4. APIs 6
   4.1. Open Game URL 6
   4.2. List Game 6
   4.3. Game Round Status 8
   4.4. Game History URL 9
   4.5. Balance 9
   4.6. Withdraw 10
   4.7. Verify Withdraw/Deposit transaction 10
   4.8. Get Withdraw/Deposit Transaction by date 11
   4.9. Sign Out 13
5. Tournament APIs 14
   5.1. Open Tournament Lobby URL 14
   5.2. Get List Tournaments 14
   5.3. Get Tournament Info 17
   5.4. Get Tournament Rank 20
6. Operator Integration APIs 22
   6.1. Authenticate Token 22
   6.2. Authenticate Username/Password (Optional) 22
   6.3. Balance 23
   6.4. Bet 24
   6.5. Settle Bet 24
   6.6. Cancel Bet 25
   6.7. Bonus Win 26
   6.8. Jackpot Win 26
   6.9. Transaction 27

6.10. Withdraw 28
6.11. Deposit 28
7. Operator Integration Tournament APIs 29
7.1. Join Tournament 29
7.2. Cancel Join Tournament 30
7.3. Win Tournament 31
8. References 32
8.1. Timestamp 32
8.2. Hash 32
8.3. Status Code 33
9. FAQ 34

1. Document History

Date	Version	Description	By
2021-Jul-07	1.23	Add **5. Tournament APIs** Add **7. Operator Integration Tournament APIs**	Customer Support
2021-May-05	1.21	Add **4.9. Sign Out** API	Customer Support
2021-Feb-18	1.20	**ID, RoundId**: Unique by member and only effect with new clients	Customer Support
2020-Nov-6	1.19	Extend the username length to **32** characters	Customer Support
2020-June-30	1.17	Get **Withdraw/Deposit** transactions	Customer Support
2020-April-20	1.15	Remove **extendedinfo**	Customer Support
2020-April-08	1.14	Add **extendedinfo** parameter for bet API	Customer Support
2020-Mar-27	1.13	The username: 4 to 20 alphanumeric characters and insensitive	Customer Support
2020-Feb-12	1.12	Add **extendedinfo** parameter for settle-bet API	Customer Support
2019-July-24	1.10	Add **balance**, **withdraw** API	Customer Support
2019-March-22	1.0	Create	Customer Support

2. API Overview

Common terms are used in the document:

- **Provider**: refers to API/System.
- **Operator**: refers to who needs to integrate with our system. Operator and Operator's system are synonymous terms in this document.

This document describes set of APIs which Operator needs to implement to integrate with Seamless Wallet API

Provider API and Operator Integration API uses **hash** to validate incoming request integrity. The hash encryption algorithm is described in section 8.2.

Some APIs are required to be **idempotent** which means making multiple identical requests has the same effect as making a single request.

Operator Integration API responses with Status integer number which indicates if the request success. The Status reference is described in section 8.3

APIs Timeout is 2.5 seconds.

3. API Login

The provider supports 2 types of login method:

- Access Token Login (suitable for Website)
- Normal Login (suitable for Mobile App)

3.1. Access Token Login

Access Token login can be used when Operator hosts the provider's games in their lobby. The flow is as following:

1. After Player logins on Operator system, Operator system passes the **Token** to open Game link:
   ```
   {gaming-url}/playGame?token={Token}&appID={AppID}&gameCode={GameCode}&language={language}&mobile={isMobile}&redirectUrl={redirectUrl}
   ```
2. Game client calls the provider API which in return calls Operator Integration API to authenticate **Token**.
3. Operator Integration API responses to validate the token together with Username, Balance
4. Provider game site shows the game.

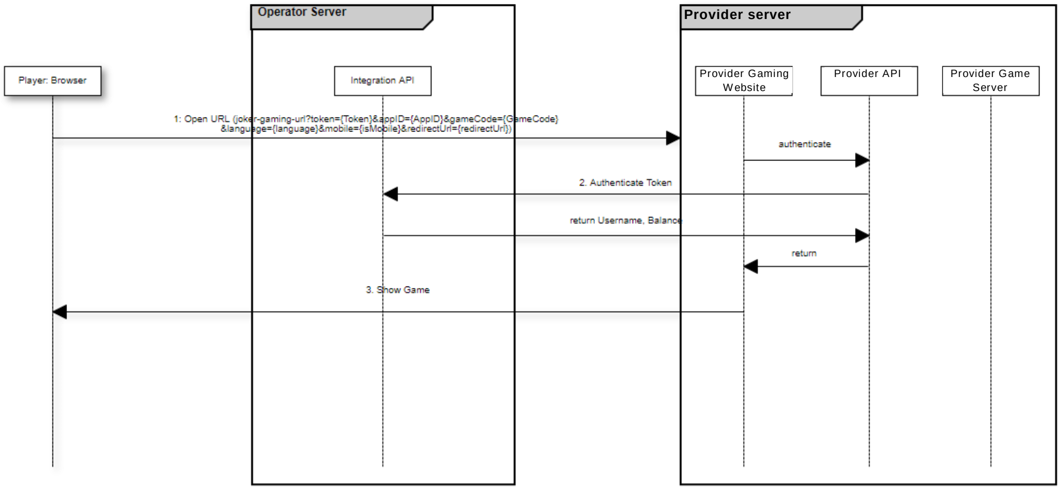

Diagram 1: Access Token Login

3.2.Normal Login

We provide normal login option which allows Operator’s players to login directly on the gaming platform (Mobile app).

1. The Player downloads the gaming platform (mobile app) and login using his username/password.
2. The game client calls API which in return calls Operator Authenticate Integration API to authenticate **username/password**.
3. Operator Integration API responses to validate the username/password together with Token, Balance.
4. Gaming App shows the game.

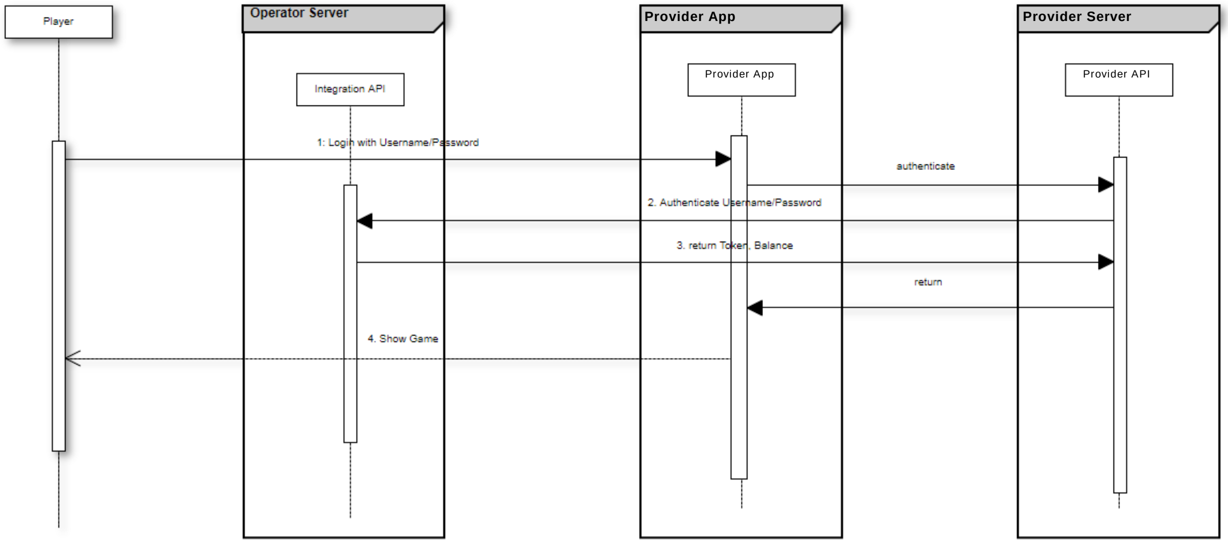

Diagram 2: Normal Login

4. APIs

4.1.Open Game URL

Request

GET `{gaming-url}/playGame?token={token}&appID={appID}&gameCode={gameCode}&language={language}&mobile={isMobile}&redirectUrl={redirectUrl}`

Property	Type	Compulsory	Description
appID	String	Y	
token	String	Y	Generated by Operator. Player is uniquely identified by **Token.**
gameCode	String	Y	Game code as listed by provider.
language	String	N	The Player's preferred language (ISO 639-1, 2 letter code)
mobile	Boolean	N	true/false
redirectUrl	String	N	The URL will be redirected when client logout

4.2.List Game

Request

POST `{api-url}/list-games`
Content-Type: application/json

```
{"AppID":"AppID","Hash":"792d2414e3b0c77e9ddea9e8aa924364","Timestamp":1528210236}
```

Property	Type	Compulsory	Description
AppID	String	Y	
Hash	String	Y	
Timestamp	Long	Y	UNIX timestamp

Response

```
{
 "Error": "0",
 "Description": "OK",
 "ListGames": [
  {
   "GameType": "Slot",
   "GameCode": "dxxsh3dfmjpio",
   "GameName": "Tai Shang Lao Jun",
      "SupportedPlatForms": "Desktop,Mobile",
      "Specials": "hot,new",
      "Order": 1,
      "DefaultWidth": 960,
      "DefaultHeight": 630,
      "Image1": "//sampledomain/gameimages/landscape/fwria11mjbrwh.png"
    }
    ....
  ]
}
```

Property	Type	Compulsory	Description

Error	String	Y	Success = **“0”**
			Failure = **“1”**

Description	String	Y	

Array of:

Property	Type	Compulsory	Description
GameCode	String	Y	Game identifier
GameName	String	Y	Game name
DefaultWidth	Integer	Y	With of game window
DefaultHeight	Integer	Y	Height of game window
GameType	String	Y	Game type e.g. Slot, Fishing
SupportedPlatForms	String	Y	
Specials	String	Y	
Order	Integer	Y	
Image1	String	Y	

4.3.Game Round Status

Request

```
POST {api-url}/game-round-status
Content-Type: application/json
```

```
{"AppID":"AppID","RoundID":"uhax9gb5upqgq","Username":"DEMO","Hash":"792d2414e3b0c77e9ddea9e8aa924364","Timestamp":1528210236}
```

Property	Type	Compulsory	Description
AppID	String	Y	
Hash	String	Y	
RoundID	String	Y	
Timestamp	Long	Y	UNIX timestamp
Username	String	Y	**Do not use this property in Hash**

Response: Content-Type: application/json

```
{
 "Error": "0",
 "Description": "OK",
 "Bets": [
  {
   "ID": "gh6gqyfdyxnx6",
   "Status": "Settled",
    "Amount": 0.10,
    "Result": 0.6
  }
  ....
 ]
}
```

Array of:

Property	Type	Compulsory	Description
ID	String	Y	BetID
Status	String	Y	Success, Cancelled, Settled

4.4.Game History URL

Request

```
POST {api-url}/game-history-url
Content-Type: application/json
{"AppID":"AppID","Username":"DEMO","GameCode":"xea1rt9gebhwy","RoundID":"uhax9gb5upqgq","Language":"en","Hash":"792d2414e3b0c77e9ddea9e8aa924364","Timestamp":1528210236}
```

Response: Content-Type: application/json

```
{
 "Error": "0",
 "Description": "OK",
 "Url":
"//History?Signature=k2Nyoas94qHkAaoIKuYAdJEnr9c%3d&Key=xkqryq3b5aeuq&Type=Game&Timestamp=1552962025093&Lang=en"
}
```

Property	Type	Compulsory	Description
URL	String	Y	

4.5.Balance

The API returns the current credit balance of the player

Request

```
POST {api-url}/balance
Content-Type: application/json

{"AppID":"AppID","Username":"DEMO,"Hash":"792d2414e3b0c77e9ddea9e8aa924364","Timestamp":1528210236}
```

Response: Content-Type: application/json

```
{
 "Error": "0",
 "Description": "OK",
 "Amount": 100.01
}
```

Property	Type	Value	Compulsory	Description
Amount	Decimal	Positive numeric	Y	

4.6.Withdraw

The API allows the Operator to trigger the deposit API (The deposit API which is used to return money back from the Provider' system to Operator's system)

Request

POST `{api-url}/withdraw`
Content-Type: application/json

```
{"AppID":"AppID","Username":"DEMO,"Hash":"792d2414e3b0c77e9ddea9e8aa924364","Timestamp":1528210236}
```

Response: Content-Type: application/json

```
{
 "Error": "0",
 "Description": "OK"
}
```

4.7. Verify Withdraw/Deposit transaction

Request

POST `{api-url}/ statement-by-request`
Content-Type: application/json

```
{"Username":"DEMO","AppID":"AppID","IDs":" 5ee99dcba3787, 5ee98746d9216",
"Timestamp":"1593144363603","Hash":"abb7cc87ea3914e46c3a9562dfe89832"}
```

Property	Type	Compulsory	Description
AppID	String	Y	
Hash	String	Y	
Username	String	Y	
Timestamp	Long	Y	UNIX timestamp
IDs	String	Y	The list requestIDs are separated by commas

Response: Content-Type: application/json

```
{
 "Error": "0",
 "Description": "OK",
 "Data": [
    {
       "Id": "5ee99dcba3787",
       "Username": "AppID.DEMO",
        "Time": "2020-06-24T15:38:55.169",
        "Amount": 20.00,
        "Type": "TopUp"
      },
      {
        "Id": "5ee98746d9216",
        "Username": "AppID.DEMO ",
        "Time": "2020-06-24T15:40:04.152",
        "Amount": -3594.90,
        "Type": "Withdraw"
      }
      ……
   ]
}
```

Property	Type	Compulsory	Description
Error	String	Y	
Description	String	Y	
Data	Array	Y	
Id	String	Y	
Username	String	Y	
Time	String	Y	
Amount	Decimal	Y	
Type	String	Y	- TopUp: Deposit - Withdraw: Withdraw

4.8.Get Withdraw/Deposit Transaction by date

Request

POST `{api-url}/ statement-by-date`
Content-Type: application/json

{"Username":"DEMO","AppID":"AppID","Date":"2020-06-24", "Timestamp":"1593498325050", "Hash":"edc6c90c32b1e148d9c2bbf4f4eeb311"}

Property	Type	Compulsory	Description
AppID	String	Y	
Hash	String	Y	
Username	String	Y	
Timestamp	Long	Y	UNIX timestamp
Date	String	Y	Date format: yyyy-MM-dd

Response: Content-Type: application/json

```
{
  "Error": "0",
  "Description": "OK",
  "Data": [
      {
         "Id": "5ee99dcba3787",
         "Username": "AppID.DEMO",
         "Time": "2020-06-24T15:38:55.169",
         "Amount": 20.00,
         "Type": "TopUp"
      },
      {
         "Id": "5ee98746d9216",
         "Username": " AppID.DEMO ",
         "Time": "2020-06-24T15:40:04.152",
         "Amount": -3594.90,
         "Type": "Withdraw"
      }
      ……
  ]
}
```

Property	Type	Compulsory	Description
Error	String	Y	
Description	String	Y	
Data	Array	Y	
Id	String	Y	
Username	String	Y	
Time	String	Y	
Amount	Decimal	Y	
Type	String	Y	- TopUp: Deposit - Withdraw: Withdraw

4.9.Sign Out

Request

```
POST {api-url}/sign-out
Content-Type: application/json

{"AppID":"AppID","Username":"Username","Hash":"792d2414e3b0c77e9ddea9e8aa924364","Timestamp":1620184560896}
```

Property	Type	Compulsory	Description
AppID	String	Y	
Username	String	Y	
Hash	String	Y	
Timestamp	Long	Y	UNIX timestamp

Response

```
{
   "Error": "0",
   "Description": "Sign out successfully."
}
```

Property	Type	Compulsory	Description

Error	String	Y	Success = **“0”**
			Failure = **“1”**

Description	String	Y	

5. Tournament APIs

5.1. Open Tournament Lobby URL

Request

```
GET {gaming-url}/playGame?token={token}&appID={appID}&gameCode=Tournament&
language={language}&mobile={isMobile}&redirectUrl={redirectUrl}
```

5.2. Get List Tournaments

Request

POST {api-url}/tournaments
Content-Type: application/json

```
{"AppID":"AppID","Status":"All","Timestamp":"1625640447662","Hash":"155786eaa5211c74b45ab
371c97b86ca"}
```

Property	Type	Compulsory	Description
AppID	String	Y	
Status	String	Y	Status has value like: - All - New - Active - Completed
Hash	String	Y	
Timestamp	Long	Y	UNIX timestamp

Response

```
{
  "Error": "0",
  "Description": "OK",
  "Data": [
    {
      "TournamentID": "xeahz7p7oag51",
      "TournamentName": "TaiShangLaoJunGW",
      "StartDate": "2021-07-07T20:00:00",
      "EndDate": "2021-07-08T22:00:00",
      "Description": "Tai Shang Lao Jun",
      "Status": "New",
      "CurrencyCode": "MYR",
      "Prize": [
        {
          "PrizeOCode": "nknny55cpitgg",
          "Amount": 10000,
          "Qty": 1,
          "Position": 1,
          "Ticket": 1,
          "Points": 1000000
        },
        {
          "PrizeOCode": "nknny55cpitgg",
          "Amount": 5000,
          "Qty": 2,
          "Position": 2,
          "Ticket": 1,
          "Points": 500000
        },
        {
          "PrizeOCode": "nknny55cpitgg",
          "Amount": 1000,
          "Qty": 10,
          "Position": 4,
          "Ticket": 1,
          "Points": 100000
        }
      ],
      "Cost": 1125,
      "RequiredTicket": 0,
      "ImageLink":
"http://dl.changxingwnet.com/tournament/assets/icon/248x248/5/TaiShangLaoJunGW.png",
      "ImageLinkSmall": "http://dl.changxingwnet.com/tournament/assets/web-
icon/320x265/1/TaiShangLaoJunGW.png",
      "Type": "total_win",
      "GameCode": "dxxsh3dfmjpio",
      "PrizePool": 39000,
      "NoOfSpins": 250,
      "TypeDescription": "Total Win",
      "TypeName": "Total Win",
      "PoolContribution": 0
    },
   …………………………..
  ]
}
```

Property	Type	Compulsory	Description

Error	String	Y	Success = **“0”**
			Failure = **“1”**

Description	String	Y	
Data	Array	Y	

Property	Type	Compulsory	Description
TournamentID	String	Y	
TournamentName	String	Y	
StartDate	DateTime	Y	
EndDate	DateTime	Y	
Description	String	Y	
Status	String	Y	
CurrencyCode	String	Y	
Cost	Decimal	Y	
RequiredTicket	Integer	Y	
ImageLink	String	Y	
ImageLinkSmall	String	Y	
Type	String	Y	
GameCode	String	Y	
PrizePool	Decimal	Y	
NoOfSpins	Integer	Y	
TypeDescription	String	Y	
TypeName	String	Y	
PoolContribution	Decimal	Y	
Prize	Array	Y	

Property	Type	Compulsory	Description
PrizeOCode	String	Y	
Amount	Decimal	Y	
Qty	Integer	Y	
Position	Integer	Y	
Ticket	Integer	Y	
Points	Long	Y	

5.3.Get Tournament Info

Request

POST {api-url}/tournament
Content-Type: application/json

```
{"AppID":"AppID","TournamentID":"xeahz7p7oag51","Timestamp":"1625640447662","Hash":"1936d0c41de7d522501b6b028f4fa178"}
```

Property	Type	Compulsory	Description
AppID	String	Y	
TournamentID	String	Y	
Hash	String	Y	
Timestamp	Long	Y	UNIX timestamp

Response

```
{
  "Error": "0",
  "Description": "OK",
  "Data": {
    "TournamentID": "xeahz7p7oag51",
    "TournamentName": "TaiShangLaoJunGW",
    "StartDate": "2021-07-07T20:00:00",
    "EndDate": "2021-07-08T22:00:00",
    "Description": "Tai Shang Lao Jun",
    "Status": "New",
    "CurrencyCode": "MYR",
    "Prize": [
      {
        "PrizeOCode": "nknny55cpitgg",
        "Amount": 10000.0,
        "Qty": 1,
        "Position": 1,
        "Ticket": 1,
        "Points": 1000000
      },
      {
        "PrizeOCode": "nknny55cpitgg",
        "Amount": 5000.0,
        "Qty": 2,
        "Position": 2,
        "Ticket": 1,
        "Points": 500000
      },
      {
        "PrizeOCode": "nknny55cpitgg",
        "Amount": 3000.0,
        "Qty": 3,
        "Position": 3,
        "Ticket": 1,
        "Points": 300000
      },
      {
        "PrizeOCode": "nknny55cpitgg",
        "Amount": 1000.0,
        "Qty": 10,
        "Position": 4,
        "Ticket": 1,
        "Points": 100000
      }
    ],
    "Cost": 1125.00,
    "RequiredTicket": 0,
    "ImageLink":
"http://dl.changxingwnet.com/tournament/assets/icon/248x248/5/TaiShangLaoJunGW.png",
    "ImageLinkSmall": "http://dl.changxingwnet.com/tournament/assets/web-
icon/320x265/1/TaiShangLaoJunGW.png",
    "Type": "total_win",
    "GameCode": "dxxsh3dfmjpio",
    "PrizePool": 39000.0,
    "NoOfSpins": 250,
    "TypeDescription": "Total Win",
    "TypeName": "Total Win",
    "PoolContribution": 0.0
  }
}
```

Property	Type	Compulsory	Description

Error	String	Y	Success = **“0”**
			Failure = **“1”**

Description	String	Y	
Data	Object	Y	

Property	Type	Compulsory	Description
TournamentID	String	Y	
TournamentName	String	Y	
StartDate	DateTime	Y	
EndDate	DateTime	Y	

Description	String	Y	
Status	String	Y	
CurrencyCode	String	Y	
Cost	Decimal	Y	
RequiredTicket	Integer	Y	
ImageLink	String	Y	
ImageLinkSmall	String	Y	
Type	String	Y	
GameCode	String	Y	
PrizePool	Decimal	Y	
NoOfSpins	Integer	Y	
TypeDescription	String	Y	
TypeName	String	Y	
PoolContribution	Decimal	Y	
Prize	Array	Y	

Property	Type	Compulsory	Description
PrizeOCode	String	Y	
Amount	Decimal	Y	
Qty	Integer	Y	
Position	Integer	Y	
Ticket	Integer	Y	
Points	Long	Y	

5.4. Get Tournament Rank

Request

POST `{api-url}/tournament-rank`
Content-Type: application/json

```
{"AppID":"AppID","TournamentID":"xeahz7p7oag51","PageIndex":0,"Size":20,"Timestamp":"1625640447662","Hash":"4492ab519ea0038e2651ea40abc236a5"}
```

Property	Type	Compulsory	Description
AppID	String	Y	
TournamentID	String	Y	
PageIndex	Integer	Y	
Size	Integer	Y	
Hash	String	Y	
Timestamp	Long	Y	UNIX timestamp

Response

```
{
 "Error": "0",
 "Description": "OK",
 "Data": [
  {
    "Username": "0000000A012",
    "DisplayName": "Tester",
    "CurrencyCode": "MYR",
    "Chance": 40770,
    "Order": 1,
    "PrizeName": "First Prize",
    "WinningAmount": 10000,
    "IsWinner": true,
    "Description": null,
    "JoinTime": "2021-06-18T16:23:23.242743",
    "Tickets": 0
  },
  ……………………………………
 ]
}
```

Property	Type	Compulsory	Description

Error	String	Y	Success = **“0”**
			Failure = **“1”**

Description	String	Y	
Data	Array	Y	

Property	Type	Compulsory	Description
Username	String	Y	
DisplayName	String	Y	
CurrencyCode	String	Y	
Chance	Integer	Y	
Order	Integer	Y	
PrizeName	String	Y	
WinningAmount	Decimal	Y	
IsWinner	Boolean	Y	
Description	String	Y	
JoinTime	Datetime	Y	
Tickets	Integer	Y	

6. Operator Integration APIs

6.1. Authenticate Token

Request

POST `{operator-integration-api-url}/authenticate-token`
Content-Type: application/x-www-form-urlencoded

Request Content:
appid=seamless&hash=1f4f020546a137427f4de3773dc04fc3&ip=128.199.64.190&timestamp=1552970614889&token=xm6rhmjciweuq

Property	Type	Compulsory	Description
token	String	Y	Operator unique identifier of player session
ip	String	Y	Player's IP
timestamp	Long	Y	UNIX timestamp
appid	String	Y	
hash	String	Y	

Response: Content-Type: application/json

```
{
    "Username": "DEMO", (4 to 32 alphanumeric characters and case-insensitive)
    "Balance": 100.01,
    "Message": "Success",
    "Status": 0
}
```

6.2. Authenticate Username/Password (Optional)

Request

POST `{operator-integration-api-url}/authenticate`
Content-Type: application/x-www-form-urlencoded

Request Content:
appid=seamless&hash=30795ed16135a0eb7711229c95670924&ip=128.199.64.190&password=abcdef&timestamp=1552976364273&username=DEMO

Property	Type	Compulsory	Description
username	String	Y	Player username **(4 to 32 alphanumeric characters and case-insensitive)**
password	String	Y	Player password
ip	String	Y	Player's IP

timestamp	Long	Y	UNIX timestamp
appid	String	Y	
hash	String	Y	

Response: Content-Type: application/json

{"Token":"xm6rhmjciweuq","Balance":100.05,"Message":"Success","Status":0}

6.3. Balance

Request

POST `{operator-integration-api-url}/balance`
Content-Type: application/x-www-form-urlencoded

Request Content:
appid=seamless&hash=45a8710d5db0aeabf4f3e3d881f732f8&timestamp=1554370334626&username=DEMO

Property	Type	Value	Compulsory	Description
appid	String		Y	
hash	String		Y	
timestamp	Long		Y	
username	String		Y	

Response: Content-Type: application/json

{"Balance":1186.11,"Message":"Success","Status":0}

6.4. Bet

Request

POST {operator-integration-api-url}/bet
Content-Type: application/x-www-form-urlencoded

Request Content:
amount=0.50&appid=seamless&gamecode=zygj7oqga9nck&hash=41399ea6eeedd294fa0dfc0b911895e7&id=B_Tb5aa0018c11e&roundid=qc9cfdk7u5wca&timestamp=1552970894238&username=DEMO

Property	Type	Value	Compulsory	Description
appid	String		Y	
hash	String		Y	
id	String		Y	BetID - Unique by member
amount	Decimal	Positive numeric	Y	
username	String		Y	
timestamp	Long		Y	
gamecode	String		Y	
roundid	String		Y	Unique by member

Response: Content-Type: application/json

{"Balance":1186.11,"Message":"Success","Status":0}

A cancel will be sent if Bet is Timeout.

6.5. Settle Bet

Request

POST {operator-integration-api-url}/settle-bet
Content-Type: application/x-www-form-urlencoded

Request Content:
amount=1.00&appid=seamless&gamecode=zygj7oqga9nck&hash=ea28cd5801d5914e027d31275f5670e4&id=S_Tb5aa0018c11f&roundid=qc9cfdk7u5wca&timestamp=1552970900488&username=DEMO&description=slot&type=main

Property	Type	Value	Compulsory	Description
appid	String		Y	
hash	String		Y	
id	String		Y	SettleID - Unique by member

amount	Decimal	Positive numeric	Y	
username	String		Y	
timestamp	Long		Y	
gamecode	String		Y	
roundid	String		Y	Unique by member
description	String		Y	
type	String		Y	

Response: Content-Type: application/json

{"Balance":1186.11,"Message":"Success","Status":0}

6.6.Cancel Bet

Request

POST `{operator-integration-api-url}/cancel-bet`
Content-Type: application/x-www-form-urlencoded

Request Content:
appid=seamless&gamecode=zygj7oqga9nck&hash=ea28cd5801d5914e027d31275f5670e4&id=C_T
b5aa0018c11f&betid=B_Tb5aa0018c11e
&roundid=qc9cfdk7u5wca&timestamp=1552970900488&username=DEMO

Property	Type	Value	Compulsory	Description
appid	String		Y	
hash	String		Y	
id	String		Y	CancelBetID - Unique by member
username	String		Y	
timestamp	Long		Y	
gamecode	String		Y	
roundid	String		Y	Unique by member
betid	String		Y	

Response: Content-Type: application/json

{"Balance":1186.11,"Message":"Success","Status":0}

6.7.Bonus Win

Request

POST `{operator-integration-api-url}/bonus-win`
Content-Type: application/x-www-form-urlencoded

Request Content:
amount=1.00&appid=seamless&gamecode=zygj7oqga9nck&hash=ea28cd5801d5914e027d31275f5670e4&id=Tb5aa0018c11f&roundid=qc9cfdk7u5wca&timestamp=1552970900488&username=DEMO&description=bonus&type=bonus

Property	Type	Value	Compulsory	Description
appid	String		Y	
hash	String		Y	
id	String		Y	Unique by member
amount	Decimal	Positive numeric	Y	
username	String		Y	
timestamp	Long		Y	
gamecode	String		Y	
roundid	String		Y	Unique by member
description	String		Y	
type	String		Y	

Response: Content-Type: application/json

{"Balance":1186.11,"Message":"Success","Status":0}

6.8.Jackpot Win

Request

POST `{operator-integration-api-url}/jackpot-win`
Content-Type: application/x-www-form-urlencoded

Request Content:
amount=1.00&appid=seamless&gamecode=zygj7oqga9nck&hash=ea28cd5801d5914e027d31275f5670e4&id=Tb5aa0018c11f&roundid=qc9cfdk7u5wca&timestamp=1552970900488&username=DEMO&description=jackpot&type=jackpot

Property	Type	Value	Compulsory	Description
appid	String		Y	
hash	String		Y	
id	String		Y	Unique by member

amount	Decimal	Positive numeric	Y	
username	String		Y	
timestamp	Long		Y	
gamecode	String		Y	**Default**: **qc8y6dypyeboy**
roundid	String		Y	Unique by member
description	String		Y	
type	String		Y	

Response: Content-Type: application/json

{"Balance":1186.11,"Message":"Success","Status":0}

6.9.Transaction

Request

POST `{operator-integration-api-url}/transaction`
Content-Type: application/x-www-form-urlencoded

Request Content:
amount=6.00&appid=seamless&description=Main&endbalance=318.00&gamecode=kk8nqm3cfwtng&hash=00b319f583ce6b2a26c2df0fe75f0d47&id=xm6qbj99cmfdq&result=4.00&roundid=xm6qbj99cmfdq&startbalance=320.00&timestamp=1555474256669&type=Main&username=DEMO

Property	Type	Value	Compulsory	Description
appid	String		Y	
hash	String		Y	
id	String		Y	Unique by member
amount	Decimal	Positive numeric	Y	
result	Decimal	Positive numeric	Y	
username	String		Y	
timestamp	Long		Y	
gamecode	String		Y	
roundid	String		Y	Unique by member
description	String		Y	
type	String		Y	
startbalance	Decimal	Positive numeric	Y	
endbalance	Decimal	Positive numeric	Y	

Response: Content-Type: application/json

{"Balance":1186.11,"Message":"Success","Status":0}

6.10. Withdraw

Request

POST `{operator-integration-api-url}/withdraw`
Content-Type: application/x-www-form-urlencoded

Request Content:
amount=1.00&appid=seamless&hash=ea28cd5801d5914e027d31275f5670e4&id=Tb5aa0018c11f&timestamp=1552970900488&username=DEMO

Property	Type	Value	Compulsory	Description
appid	String		Y	
hash	String		Y	
id	String		Y	Unique by member
amount	Decimal	Positive numeric	Y	
username	String		Y	
timestamp	Long		Y	

Response: Content-Type: application/json

{"Balance":1186.11,"Message":"Success","Status":0}

6.11. Deposit

Request

POST `{operator-integration-api-url}/deposit`
Content-Type: application/x-www-form-urlencoded

Request Content:
amount=1.00&appid=seamless&hash=ea28cd5801d5914e027d31275f5670e4&id=Tb5aa0018c11f&timestamp=1552970900488&username=DEMO

Property	Type	Value	Compulsory	Description
appid	String		Y	
hash	String		Y	
id	String		Y	Unique by member
amount	Decimal	Positive numeric	Y	
username	String		Y	
timestamp	Long		Y	

Response: Content-Type: application/json {"Balance":1186.11,"Message":"Success","Status":0}

7. Operator Integration Tournament APIs

7.1.Join Tournament

Join Tournament is called when the player joins a tournament. Operator is expected to deduct the balance of player and record the transaction

Request

POST {operator-integration-api-url}/join-tournament
Content-Type: application/x-www-form-urlencoded

Request Content:
amount=10.00&appid=seamless&extendedinfo=%7B%22PoolContribution%22%3A2.00%7D&hash=c58166e6524e28cbdb47037133a8f83a&id=jxeahin3up4q51&timestamp=1625644814173&tournamentid=xeahin3up4q51&username=DEMO

Property	Type	Value	Compulsory	Description
appid	String		Y	
hash	String		Y	
id	String		Y	Unique by member
tournamentid	String		Y	
amount	Decimal	Positive numeric	Y	
username	String		Y	
timestamp	Long		Y	
extendedinfo	String		Y	- **Do not use this property in Hash** - Is JSON string. Client can deserialize this value to object if need to consume. For example: JSON.parse(value)

Response: Content-Type: application/json {"Balance":1186.11,"Message":"Success","Status":0}

7.2.Cancel Join Tournament

Operator is expected to refund the balance of player (if the balance is deducted when player joins the tournament) and to remove transaction if there is

Request

POST `{operator-integration-api-url}/cancel-join-tournament`
Content-Type: application/x-www-form-urlencoded

Request Content:
appid=seamless&extendedinfo=%7B%7D&hash=9bbf0aacc55559c922343588229dd905&id=cxeahin3up4q51&jointournamentid=xeahin3up4q51&timestamp=1625645436243&username=DEMO

Property	Type	Value	Compulsory	Description
appid	String		Y	
hash	String		Y	
id	String		Y	Unique by member
jointournamentid	String		Y	
username	String		Y	
timestamp	Long		Y	
`extendedinfo`	String		Y	**- Do not use this property in Hash** - Is JSON string. Client can deserialize this value to object if need to consume. For example: JSON.parse(value)

Response: Content-Type: application/json {"Balance":1186.11,"Message":"Success","Status":0}

7.3. Win Tournament

Win Tournament is called when the tournament is finished and player wins of the tournament. Operator is expected to add the balance of player and record the transaction

Request

POST `{operator-integration-api-url}/win-tournament`
Content-Type: application/x-www-form-urlencoded

Request Content:
amount=5000.00&appid=seamless&extendedinfo=%7B%22PoolContribution%22%3A2.00%7D&hash=c9d19dcb9cea559c3d111c1779002534&id=wxeahin3up4q51&timestamp=1625646194210&tournamentid=xeahin3up4q51&username=DEMO

Property	Type	Value	Compulsory	Description
appid	String		Y	
hash	String		Y	
id	String		Y	Unique by member
tournamentid	String		Y	
username	String		Y	
amount	Decimal	Positive numeric	Y	
timestamp	Long		Y	
`extendedinfo`	String		Y	**- Do not use this property in Hash** - Is JSON string. Client can deserialize this value to object if need to consume. For example: JSON.parse(value)

Response: Content-Type: application/json {"Balance":1186.11,"Message":"Success","Status":0}

8. References

8.1.Timestamp

The timestamp is used as part of request to verify integrity. If the request is much later than its timestamp, Operator should reject the request.

Timestamp is defined as the number of **miliseconds** elapsed since midnight proleptic Coordinated Universal Time (UTC) of January 1, 1970.

Following sample code in C# demonstrating how to get current timestamp:

```
public static readonly DateTime UnixEpoch = new DateTime(1970, 1, 1, 0, 0, 0, 0,
DateTimeKind.Local);
public static long GetCurrentTimestamp()
{
        return (long)DateTime.UtcNow.Subtract(UnixEpoch).TotalMilliseconds;
}
```

8.2.Hash

The provider requested with **MD5** encrypted signature. For example, if request is:

{"key1": "value1", "key3": "value3", "key2": "value2", **"key4": ""** }

Step 1: Construct raw signature string in query string format and append with secret key. Keys are in lower case and are **sorted** in alphabetic order. Those key pairs with no value are omitted. **The decimal numbers are two places decimal numeric formatted e.g. 1.00**

```
String rawData = key1=value1&key2=value2&key3=value3secretKey
```

Example:

```
AppID: TXA2M
Timestamp: 1561025342810
Amount: 1000.0  (or 1000 or 1000.00)
ID: xe6jrzgntw1pc
Username: DEMO
SecretKey: 123456

==> String rawData =
amount=1000.00&appid=TXA2M&id=xe6jrzgntw1pc&timestamp=1561025342810&username=DEMO1234
56
==> Hash = f38c4df2641a9c40129ae039a9ef8def
```

Step 2: Use secret key with **MD5** encryption algorithm to encrypt data.

```
public static string GetMD5Signature(string rawData)
{
        var md5Content = string.Empty;
        using (MD5 md5Hash = MD5.Create())
        {
           md5Content = SignatureHelper.GetMd5Hash(md5Hash, rawData);
        }
        return md5Content;
 }



private static string GetMd5Hash(MD5 md5Hash, string rawData)
 {
        // Convert the input string to a byte array and compute the hash.
        byte[] data = md5Hash.ComputeHash(Encoding.UTF8.GetBytes(rawData));

        // Create a new Stringbuilder to collect the bytes
        // and create a string.
        var sBuilder = new StringBuilder();

        // Loop through each byte of the hashed data
        // and format each one as a hexadecimal string.
        for (int i = 0; i < data.Length; i++)
        {
           sBuilder.Append(data[i].ToString("x2"));
        }

        // Return the hexadecimal string.
        return sBuilder.ToString();
 }
```

Note: the signature may contain some special character and it should be converted into escaped representation used for query string.

8.3.Status Code

Status	Description	Applicable APIs
0	Success	All
1	IP not allowed	All
2	Invalid AppID	All
3	Invalid token	Authenticate-Token
4	Invalid parameters	All
5	Invalid signature	All
6	Invalid timestamp	All

7	Invalid Username or Password	Authenticate
100	Insufficient fund	Bet, Withdraw
201	Transaction is being processed	All
999	Server under maintenance	All
1000	Other	All

9. FAQ

1. **How "Bet" is associated with "Settle"?**

   A Settle can be for multiple Bet. Bet and Settle are associated by RoundID.

2. **When to use the "Bonus Win" API?**

   The Bonus can be thought as Settle without Bet. Third party should add balance to player when receive the event.

3. **When to use the "Jackpot Win" API?**

   The Jackpot can be thought as Settle without Bet. Third party should add balance to player when receive the event.

4. **Can “Cancel” be received after “Settle”?**

   Yes. There is case that Cancel will be sent after Settle. In this case, the balance should be added back to player. Third party should adjust transaction.

5. **When to use the “Withdraw” API?**

   This API is used for to player transfer money to play game Fish.

6. **When to use the “Deposit” API?**

   This API is used for to return money back.

7. **When to use the “Transaction” API?**

   This API is used for to send bet detail grouping transaction (Fish) in which the bet detail is not sent when Bet. The API has no impact on the player balance.

8. **When to use the “Transaction is being processed” status code?**

   When Provider sends the first request to Third party but it processed longer than the timeout, Provider will re-send again.

   a. The first request is being processed -> the second request should be returned with the status **201**
   b. The first request is completed -> the second request should be returned with the correct status (which makes the call idempotent).