

FastApi

FastAPI provides a simple and secure way to integrate with third-party platforms for account creation, deposits, and withdrawals.

Preparations

The integrator must obtain the following authorization details:

- API Server Domain.
- appid: API integration identification ID.
- appsecret: API authentication private key.

The integrator must provide their **server IP address** to be added to the API access whitelist.

Request Format

Unless specified otherwise, API requests should use **HTTP POST** with application/x-www-form-urlencoded format.

- appid: API integration identification ID.
- timestamp: Request timestamp (milliseconds)
- sign: Signature generated by the signing algorithm

Example JSON Request:

```
json
{
  "appid": "h63inikngg2qyoii",
  "timestamp": 1701401242061,
  "sign": "c685901b15253d820dc1bc062d21d4f1"
}
```

Response Format

All API responses follow this format:

```
json
{
  "msg": "message",
  "code": 0,
  "data": {object}
}
```

Request API Signature:

This signature is used when the client sends requests to the FastAPI server.

Steps:

1. Exclude the sign field from parameters.
2. Sort the remaining fields in ascending order by **parameter name**.
3. Format each parameter as key=value, then concatenate them using the & character to form the parameter string.
4. Append the given **appsecret** to the end of the concatenated parameter string.
5. Generate the signature by applying the **MD5 hash** algorithm to the resulting string.
6. For **POST requests (with JSON)** and **GET requests (with URL parameters)**, apply the same sorting and signing process.
7. The server will **only accept requests within 5 minutes** of the provided timestamp.

Signature Code Example:

```
function sign(array $data, string $appSecret)
{
    unset($data['sign']);
    $params = array_map(function ($value) {
        if (is_array($value)) {
            $value = json_encode($value);
        } elseif (is_bool($value)) {
            $value = var_export($value, true);
        }
        return $value;
    }, $data);
    ksort($params);
    $strArr = [];
    foreach ($params as $key => $value) {
        $strArr[] = $key . '=' . $value;
    }
    return md5(implode('&', $strArr) . $appSecret);
}
```

Response Code

Description of the Code field values in API responses:

Code	Meaning
200	Success
1	New User Is Created
2	User Does Not Exist
3	Parameter Error
4	Invalid Signature
5	Agent Ban
6	Account length error
7	Account format error
8	Password length error

9	Password format error
10	Requestid Used
11	Unknown Database Error
12	User Already Exist
13	Top Up Fail
14	Insufficient Credit
15	Withdrawal Failed
16	Get Balance Failed
17	Operations are Not Allowed In The Game
18	System Is Under Maintenance
19	The Requested Address Does Not Exist
20	Password error
21	Agent Name Or Password error
22	Platform Not Configured

AES Decryption of appsecret_encrypted

1. Perform base64_decode on the appsecret_encrypted string returned by the agent login API.
2. Extract the first 16 characters as the initialization vector (IV), and use the remaining characters starting from the 17th as the data to be AES decrypted.
3. Convert the agent password to lowercase, then apply MD5 hashing twice to generate the AES decryption key.
4. Use the AES-256-CBC algorithm to decrypt the data using the key and IV to obtain the appsecret.

Sample AES Decryption Code:

```
function aesDecrypt($data, $key) {
    $data = base64_decode($data);
    $iv = substr($data, 0, 16); // Extract the first 16 bytes as IV
    $encrypted = substr($data, 16); // The rest is the encrypted data
    return openssl_decrypt($encrypted, 'AES-256-CBC', $key, OPENSSL_RAW_DATA, $iv);
}
```

1. Agent Login

Path: /fast/agent/login

Method: POST

The appsecret_encrypted returned by the API that has been encrypted using the AES algorithm. In order to use the actual appsecret, the client must perform AES decryption on the specified decryption process.

Headers:

Parameter Name	Parameter Value	Required
Content-Type	application/x-www-form-urlencoded	Yes

Body:

Parameter Name	Type	Required	Remark
----------------	------	----------	--------

requestid	text	Yes	Unique request ID (up to 64 alphanumeric characters)
timestamp	text	Yes	Request timestamp (milliseconds)
account	text	Yes	Agent account
passwd	text	Yes	Agent password
sign	text	Yes	sign

Response Data:

Parameter Name	Type	Required	Remark
code	integer	Yes	Status Code
message	string	Yes	Description
data	object	Yes	
└─ balance	number	Yes	Agent balance
└─ appid	string	Yes	appid
└─ appsecret_encrypted	string	Yes	The appsecret after AES encryption

2. Create Player

Path: /fast/user/create

Method: POST

Headers:

Parameter Name	Parameter Value	Required
Content-Type	application/x-www-form-urlencoded	Yes

Body:

Parameter Name	Type	Required	Remark
requestid	text	Yes	Unique request ID (up to 64 alphanumeric characters)
appid	text	Yes	appid
timestamp	text	Yes	Request timestamp (milliseconds)
sign	text	Yes	sign
account	text	Yes	Player account(not include prefix, 3–16 characters, only letters/numbers)
passwd	text	Yes	Player password (6–16 characters, must include letters and numbers; allowed symbols: !@#\$()%^/.,)

Response Data:

Parameter Name	Type	Required	Remark
code	integer	Yes	Status Code
message	string	Yes	Description
data	object	Yes	
└─ full_account	string	Yes	Full account name with prefix

3. Deposit

Path: /fast/user/deposit

Method: POST

Headers:

Parameter Name	Parameter Value	Required
Content-Type	application/x-www-form-urlencoded	Yes

Body:

Parameter Name	Type	Required	Remark
requestid	text	Yes	Unique request ID (up to 64 alphanumeric characters)
appid	text	Yes	appid
timestamp	text	Yes	Request timestamp (milliseconds)
sign	text	Yes	sign
account	text	Yes	Player account (not include prefix)
amount	text	Yes	e.g., 100.55, up to 2 decimal places

Response Data:

Parameter Name	Type	Required	Remark
code	integer	Yes	Status Code
message	string	Yes	Description
data	object	Yes	
— balance	number	Yes	Updated balance
— order_num	string	Yes	Order number
— requestid	string	Yes	Unique request ID
— time	integer	Yes	Timestamp of the operation

4. Withdrawal

Path: /fast/user/withdrawal

Method: POST

Headers:

Parameter Name	Parameter Value	Required
Content-Type	application/x-www-form-urlencoded	Yes

Body:

Parameter Name	Type	Required	Remark
requestid	text	Yes	Unique request ID (up to 64 alphanumeric characters)
appid	text	Yes	Appid
timestamp	text	Yes	Request timestamp (milliseconds)
sign	text	Yes	sign
account	text	Yes	Player account (not include prefix)
amount	text	Yes	e.g., 100.55, up to 2 decimal places

Response Data:

Parameter Name	Type	Required	Remark
code	integer	Yes	Status Code
message	string	Yes	Description

data	object	Yes	
— balance	number	Yes	Updated balance
— order_num	string	Yes	Order number

5. Get Balance

Path: /fast/user/balance

Method: POST

Headers:

Parameter Name	Parameter Value	Required
Content-Type	application/x-www-form-urlencoded	Yes

Body:

Parameter Name	Type	Required	Remark
requestid	text	Yes	Unique request ID (up to 64 alphanumeric characters)
appid	text	Yes	appid
timestamp	text	Yes	Request timestamp (milliseconds)
sign	text	Yes	sign
account	text	Yes	Player account (not include prefix)

Response Data

Parameter Name	Type	Required	Remark
code	integer	Yes	Status Code
message	string	Yes	Description
data	object	Yes	
— balance	number	Yes	Player's updated balance

6. Get Balance (With Password)

Path: /fast/user/balanceWithPasswd

Method: POST

Headers:

Parameter Name	Parameter Value	Required
Content-Type	application/x-www-form-urlencoded	Yes

Body:

Parameter Name	Type	Required	Remark
requestid	text	Yes	Unique request ID (up to 64 alphanumeric characters)
appid	text	Yes	appid
timestamp	text	Yes	Request timestamp (milliseconds)
sign	text	Yes	sign
account	text	Yes	Player account (not include prefix)
passwd	text	Yes	Player password

Response Data

Parameter Name	Type	Required	Remark
code	integer	Yes	Status Code
message	string	Yes	Description
data	object	Yes	
— balance	number	Yes	Player's updated balance

7. Change Password

Path: /fast/user/updatePasswd

Method: POST

Headers:

Parameter Name	Parameter Value	Required
Content-Type	application/x-www-form-urlencoded	Yes

Body:

Parameter Name	Type	Required	Remark
requestid	text	Yes	Unique request ID (up to 64 alphanumeric characters)
appid	text	Yes	appid
timestamp	text	Yes	Request timestamp (milliseconds)
sign	text	Yes	sign
account	text	Yes	Player account (not include prefix, 3–16 characters, only letters/numbers)
passwd	text	Yes	(6–16 characters, must include letters and numbers; allowed symbols: !@#\$(%^/.,)
new_passwd	text	Yes	New password

Response Data

Parameter Name	Type	Required	Remark
code	integer	Yes	Status Code
message	string	Yes	Description

8. Get Trade List

Path: /fast/user/tradeList

Method: POST

Headers:

Parameter Name	Parameter Value	Required
Content-Type	application/x-www-form-urlencoded	Yes

Body:

Parameter Name	Type	Required	Remark
requestid	text	Yes	Unique request ID (up to 64 alphanumeric characters)
appid	text	Yes	appid
timestamp	text	Yes	Request timestamp (milliseconds)
sign	text	Yes	sign
account	text	Yes	Player account (not include prefix)

start_date	text	Yes	Start date (ex: 2025-06-01)
end_date	text	Yes	End date (ex: 2025-06-11)
page	text	Yes	Page number (pass 0 for first page)
page_num	text	Yes	Number of records per page (ex: 20)

Response Data

Parameter Name	Type	Required	Remark
code	integer	Yes	Status Code
message	string	Yes	Description
data	object	Yes	
— total	integer	Yes	Total record
— pages	boolean	Yes	Total page
— list	object []	Yes	Data list
— order_num	string	Yes	Order number
— start_score	number	Yes	Score before operation
— score	number	Yes	Operation score (+ deposit, - withdrawal)
— time	integer	Yes	Times of the operation

9. Get Gamelog List

Path: /fast/user/gameLogList

Method: POST

Note: Retrieve up to the latest 1,000 game records

Headers:

Parameter Name	Parameter Value	Required
Content-Type	application/x-www-form-urlencoded	Yes

Body:

Parameter Name	Type	Required	Remark
requestid	text	Yes	Unique request ID (up to 64 alphanumeric characters)
appid	text	Yes	appid
timestamp	text	Yes	Request timestamp (milliseconds)
sign	text	Yes	sign
account	text	Yes	Player account (not include prefix)

Response Data

Parameter Name	Type	Required	Remark
code	integer	Yes	Status Code
message	string	Yes	Description
data	object	Yes	
— list	object []	Yes	Data List
— game_id	integer	Yes	Game ID
— game_name	string	Yes	Game Name
— start_score	string	Yes	Score before operation
— end_score	string	Yes	Operation score (+ deposit, - withdrawal)
— pay	string	Yes	Bet amount
— win	string	Yes	Win amount
— time	integer	Yes	Times of the operation

☐ — unqleid	string	Yes	Unique ID
------------------------------------	--------	-----	-----------