



API lottery game interface

description

Revision record:

| Revision time | Modifier | description |
|---------------|----------|--|
| 2018-03-12 | Daniel | First draft |
| 2020-06-15 | Daniel | Test address: http://api.lffhbw.top/ |
| 2020-08-10 | Daniel | <u>Quota interface</u> , <u>Add return value description</u> <u>Single bet interface</u> . Newly added status-2 canceled <u>Merchant operation</u> <u>New operation type type=12</u> |
| 2020-08-14 | Daniel | <u>AddRollback interface</u> To facilitate the rollback of abnormal amounts <u>Single bet interface</u> <u>Addedserialnumber parameter</u> |
| 2020-08-27 | Daniel | <u>Login interface</u> <u>New token parameter</u> <u>Add2.9.1-2.9.6 Merchant interface (application/json)</u> |
| 2021-04-22 | Daniel | <u>Add Game Type Description</u> <u>Add Game Result Description</u> |

Table of Contents

| | |
|---|-----------|
| Document description | 4 |
| Access platform interface description | 5 |
| 2.1 Interface definition | 5 |
| 2.2 Login interface (/api/login) | 5 |
| 2.3 Kick out the game (/api/kickout) | 7 |
| 2.4 Create user (/api/register) | 7 |
| 2.5 Modify user basic information (/api/user/update) | 8 |
| 2.6 Get basic user information (/api/user/info) | 9 |
| 2.7 User transfer form | 10 |
| 2.7.1 Get user balance (/api/user/balance) | 11 |
| 2.7.2 User deposits and withdrawals (/api/user/dw) | 12 |
| 2.7.3 Get deposit and withdrawal record interface (/api/user/trade) | 12 |
| 2.7.4 Quota list interface (/api/quota/list) | 14 |
| 2.7.5 Quota setting interface (/api/quota/set) | 15 |
| 2.8 Single wallet form for merchant users (implemented by merchant access party) | 16 |
| 2.8.1 Message type description | 16 |
| 2.8.2 Merchant user amount query interface (/api/balance) | 17 |
| 2.8.3 Merchant user deduction interface (/api/deduct) | 18 |
| 2.8.4 Merchant user deposit interface (/api/deposit) | 20 |
| 2.8.5 Merchant user rollback interface (/api/rollback) | 22 |
| 2.9 Merchant user single wallet JSON format (implemented by merchant access party) | 24 |
| 2.9.1 Message type description | 24 |
| 2.9.2 Merchant user amount query interface (/api/balance) | 25 |
| 2.9.3 Merchant user deduction interface (/api/deduct) | 26 |
| 2.9.4 Merchant user deposit interface (/api/deposit) | 28 |
| 2.9.5 Merchant user rollback interface (/api/rollback) | 30 |
| 2.9.6 Merchant bets active push interface (/api/push/bets) | 32 |
| 2.10 Get game bet slip interface (/api/data/getbets) | 33 |
| 2.11 Callback processing betting status (/api/data/removebets) | 34 |
| 2.12 Get the game list interface (/api/data/getgames) | 35 |
| 2.13 Get the game history result interface (/api/record/game/history) | 36 |

| | |
|--|-----------|
| 2.14 Get bet statistics interface (/api/record/bets/stat) | 37 |
| 2.15 Get the detailed interface of bets (/api/record/bets/detail) | 38 |
| 2.16 Betting follow interface (/api/user/bets/follow) | 39 |
| 2.17 Account details query interface (/api/record/account/detail) | 40 |
| 2.18 Re-settlement interface (/api/user/bets/settle) | 41 |
| 3. Appendix reference | 42 |
| 3.1 Error code | 42 |
| 3.2 Language code | 43 |
| 3.3 Game Type | 43 |
| 3.4 Game Result Code | 43 |
| 3.5 Account operation type | 44 |
| 3.6 Currency code | 44 |

1. Document description

The interface name is defined by the document and is case sensitive:

The interface parameter names are uniformly lowercase, case sensitive, and interface parameters and returns are uniformly coded in UTF-8:

The interfaces are all in the form of http://host:port/api/1
host (122.152.173.100) is the host address, port is the port number (the example is temporarily unavailable):

api is the prefix that must be added to the interface

"?" is the request interface name:

Description of param: indicates interface parameters, specifically the field names defined in each interface. Interface parameters are usually first letter lowercase, refer to the definition of param in each interface:

The parameter list of the interface consists of "param=value" in the form of "param=value" linked by "&":

"Is it necessary" in the interface document, Y means the interface must be passed, and N means not necessary:

About the description returned by each interface in the interface description:

Each interface has been returned in JSON format, and the specific interface document shall prevail, in the form:

Error message: {"result": "-1", "errid": 40013, "desc": "invalid appid"}

, Where result represents the status, negative numbers are abnormal, positive numbers are each meaning, desc is the request description, 0 is successful, and -1 is abnormal. Specific reference to the return value of each interface

The lowercase of the uniform parameters of the returned results must be strictly in accordance with the specifications for development and reading (the actual results shall prevail)

prefix is the company prefix, assigned by the game service

Encryption signature description:

The signature method is MD5, and the default encoding is utf-8.

Sort all parameters (empty values do not participate) from small to large according to ASCII, and after sorting, separate the links with &

(stringA="appid=wxd930ea5d5a258f4f&body=test&device_info=1000&mch_id=10000100&nonce_str=ibuaiVcKdpRkhJA";).

Put the agreed MD5 key at the end:

(stringSignTemp=stringA+"&key=192006250b4c09247ec02edce69f6a2d")

Participate in the signature (MD5 (stringSignTemp)).

2. Access platform interface description

2.1 Interface definition

2.2 Login interface (/api/login)

| | | | |
|---|--|--------------|-----|
| Interface name: /login | Request method: POST | | |
| Example URL: http://api.uobpc.cn/api/login | | | |
| Input data format: appid username ... sign | | | |
| Field name | description | Numeric type | Y/N |
| appid | Assign application ID | String | Y |
| username | log-in name | String | Y |
| nickname | Nickname (not transmitted randomly generated by the system) | String | N |
| state | Status (1 open, 0 prohibited, -1 locked, -2 prohibited betting, -3 exceeds to win, -4 exceeds to lose) | byte | N |
| commratio | Accounted for into the field (defined by the company, to avoid temporary modification of the proportion of commission) The host id can be transmitted based on this (ID10001_0.5) | String | N |
| quotas | Quota field ("1, 2, 3" separated by comma) reference Quota interface | String | N |
| iscreate | Whether to create an account (1, yes, 0 no) | byte | Y |
| clienttype | Mobile phone identification (1 pc, 2 mobile phones, 3 ipad) | byte | Y |
| language | Language identification (reference Language code) | byte | Y |
| portrait | profile picture | String | N |
| currency | Currency code (CNH, USD, international universal code) | String | N |
| returnurl | Return address (used to return to the original website after entering the game) | String | N |
| gt | game type(4/Baccarat, 5/Roulette, 10/Dragon Tiger, 11/Sic Bo) | Byte | N |
| gid | Game id, select a single game to enter | Integer | N |
| ist | Trial logo (ist=1 Trial) Try to play 1000 gold coins without accounting | Byte | N |

| | | | |
|--------------|------------------------------|--------|---|
| token | Merchant sessionToken | String | N |
| sign | Encryption key | String | Y |

| |
|--|
| return Web success example: {"result":0,"openurl":"http://api.uobpc.cn/api/opengame.jsp?code=2016060611192726701102438&language=1","desc":"OK!"} Examples of errors: {"result":-1,"desc":"error!"} |
|--|

| Field name | description | Numeric type | Do you have to |
|----------------|--|--------------|----------------|
| desc | description | String | Y |
| result | Result information (success>=0, error<0, less than zero is abnormal) | Int | Y |
| openurl | Enter the game link | String | N |

2.3 Kick out the game (/api/kickout)

| | | | |
|---|--|--------------|----------------|
| Interface name: /kickout | Request method: POST | | |
| Example URL: http://api.uobpc.cn/api/kickout | | | |
| Input data format: appid username ... sign | | | |
| Field name | description | Numeric type | Y/N |
| appid | Assign application ID | String | Y |
| username | log-in name | String | Y |
| sign | Encryption key | String | Y |
| <pre>return</pre> | | | |
| Web success example: { "result": "0", "desc": "OK" } | | | |
| Examples of errors: { "desc": "error!", "result": -1 } | | | |
| Field name | description | Numeric type | Do you have to |
| desc | description | String | Y |
| result | Result information (0 success, failure <0) | Int | Y |

2.4 Create user (/api/register)

| | | | |
|---|--|--------------|----------------|
| Interface name: register | Request method: post | | |
| Example URL: http://api.uobpc.cn/api/register | | | |
| Input data format: appid username ... sign | | | |
| Field name | description | Numeric type | Y/N |
| appid | Assign application ID | String | Y |
| username | log-in name | String | Y |
| nickname | Nickname (not transmitted randomly generated by the system) | String | N |
| state | Status (1 open, 0 prohibited, -1 locked, -2 prohibited betting, -3 exceeds to win, -4 exceeds to lose) | byte | N |
| commratio | Commission accounted for (defined by the company, to avoid temporary modification of commission to steal commission) | String | N |
| quotas | Quota field ("1, 2, 3" separated by comma) reference Quota interface | String | N |
| returnurl | Return address (used to return to the original website after entering the game) | String | N |
| sign | Encryption key | String | Y |
| return | | | |
| Successful example: { "result": "0", "desc": "OK"} | | | |
| Examples of errors: { "result": -4, "desc": "signature error!"} | | | |
| Field name | description | Numeric type | Do you have to |
| desc | description | String | Y |
| result | Result information (0 success, -1 failure) | Int | Y |

2.5 Modify user basic information

(/api/user/update)

| Interface name: update | Request method: post | | |
|---|--|--------------|-----|
| Example URL: | | | |
| http://api.uobpc.cn/api/user/update | | | |
| Input data format: appid username ... sign | | | |
| Field name | description | Numeric type | Y/N |
| appid | Assign application ID | String | Y |
| username | log-in name | String | Y |
| nickname | Nickname (not transmitted randomly generated by the system) | String | N |
| state | Status (0 open, -1 prohibited, -2 locked) | byte | N |
| betstate | Bet status (0 open, -1 prohibited, -2 exceeds to win, -3 exceeds to lose":) | byte | N |
| commratio | Commission accounted for (defined by the company, to avoid temporary modification of commission to steal commission) | String | N |
| quotas | Quota field ("1, 2, 3" separated by comma) reference Quota interface | String | N |
| sign | Encryption key | String | Y |

| return | | | |
|---|--|--------------|----------------|
| Successful example: { "result": "0", "desc": "OK" } | | | |
| Examples of errors: { "result": -4, "desc": "signature error!" } | | | |
| | | | |
| Field name | description | Numeric type | Do you have to |
| desc | description | String | Y |
| result | Result information (0 success, failure <0) | Int | Y |

2.6 Get basic user information (/api/user/info)

| | |
|---|-----------------------|
| Interface name: /user/info | Request method: get |
| Example URL: http://api.uobpc.cn/api/user/info | |
| Input data format: appid username ... sign | |
| Field name | description |
| appid | Assign application ID |
| username | log-in name |
| sign | Encryption key |

| | |
|--|---|
| return | |
| Successful example: { "result":0,"userdata":{"betstate":0,"quota":"A,B","commratio":"11111","state":0}} | |
| Examples of errors: { "result":-4,"desc":"signature error!"} | |
| Field name | description |
| result | Result information (0 success, failure <0) |
| userdata | |
| state | Status (0 open, -1 prohibited, -2 locked) |
| betstate | Bet status (0 open, -1 prohibited, -2 exceeds to win, -3 exceeds to lose") |
| commratio | Commission accounted for (defined by the company, to avoid temporary modification of commission to steal commission) |
| quota | Limit plan (see appendix) |
| state | Status (0 open, -1 prohibited, -2 locked) |

2.7 User transfer form

2.7.1 Get user balance (/api/user/balance)

| Interface name: /user/balance | Request method: get | | |
|---|--|--------------|----------------|
| Example URL: http://api.uobpc.cn/api/user/balance | | | |
| Input data format: appid username ... sign | | | |
| Field name | description | Numeric type | Y/N |
| appid | Assign application ID | String | Y |
| username | log-in name | String | Y |
| sign | Encryption key | String | Y |
| return | | | |
| Successful example: { "result":0,"balance":100} | | | |
| Examples of errors: { "result":-4,"desc":"signature error!"} | | | |
| Field name | description | Numeric type | Do you have to |
| result | Result information (0 success, failure <0) | Int | Y |
| balance | Balance | Float | Y |
| isonline | Is online | Boolean | Y |
| isplaying | Is there a bet | Boolean | Y |

2.7.2 User deposits and withdrawals (/api/user/dw)

| Interface name: /user/dw | Request method: post | | | | | |
|---|--|--------------|----------------|--|--|--|
| Example URL: http://api.uobpc.cn/api/user/dw | | | | | | |
| Input data format: appid username ... sign | | | | | | |
| Field name | description | Numeric type | Do you have to | | | |
| appid | Assign application ID | String | Y | | | |
| username | log-in name | String | Y | | | |
| tradeno | Order number (when not transmitted, the system will generate by default) | long | N | | | |
| amount | Amount (negative withdrawal, positive deposit) | Double | Y | | | |
| sign | Encryption key | String | Y | | | |

| return | | | | | | |
|--|--|--------------|----------------|--|--|--|
| Successful example: { "result": "0", "desc": "OK", "balance": "20000" } | | | | | | |
| Examples of errors: { "result": -4, "desc": "signature error!" } | | | | | | |
| | | | | | | |
| Field name | description | Numeric type | Do you have to | | | |
| desc | description | String | Y | | | |
| result | Result information (0 success, failure <0) | Int | Y | | | |
| orderno | order number | long | Y | | | |
| balance | User balance | double | Y | | | |

2.7.3 Get deposit and withdrawal record interface

(/api/user/trade)

| Interface name: /user/trade is implemented by the merchant company | | | | | | |
|---|-----------------------|--------------|----------------|--|--|--|
| Example URL: http://api.uobpc.cn/api/user/trade | | | | | | |
| Input data format: appid ... sign | | | | | | |
| | | | | | | |
| Field name | description | Numeric type | Do you have to | | | |
| appid | Assign application ID | String | Y | | | |
| username | log-in name | String | Y | | | |
| tradeno | order number | String | N | | | |

| | | | |
|------------------|-------------------|---------------|---|
| begintime | Start time | Int (seconds) | Y |
| endtime | End Time | Int (seconds) | Y |
| pageindex | Start page | int | Y |
| pagesize | Page size | int | Y |
| sign | Encryption key | String | Y |

| return | | | |
|--|--|--------------|----------------|
| Web success example: | | | |
| <code>{"result":0,"data":[],"total":1}</code> | | | |
| Examples of errors: | | | |
| <code>{"result":-1,"data":[],"total":0}</code> | | | |
| Field name | description | Numeric type | Do you have to |
| result | Result information (0 success, -1 failure) | Int | Y |
| total | Total number of queries | Int | Y |
| data | Return collection | JSONArray | Y |
| ID | Record ID | Long | N |
| type | Record type (7 deposits, 8 withdrawals) | byte | N |
| tradeno | Record number | long | N |
| payway | Payment method (11 interface format) | long | N |
| amount | Amount | double | N |
| state | Status (-1 failed, 1 application, 2 processing, 11 successful) | byte | N |
| desc | Remarks | String | N |
| auditid | Operator id | String | N |
| updatetime | Change the time | int | N |
| createtime | Creation time | int | N |

2.7.4 Quota list interface (/api/quota/list)

| Interface name: /api/user/quota/list | | | |
|---|---|--------------|----------------|
| Example URL: http://api.uobpc.cn/api/quota/list?appid=*&... | | | |
| Input data format: appid ... sign | | | |
| appid | Assign application ID | String | Y |
| notifyid | Message number, the message number of each notification is different | String | N |
| sign | Encryption key | String | Y |
| return | | | |
| Successful example: { "result":0,"array": [{"ID":1,"up":75.0,"title":"0.5-75.0","down":0.5}, {"ID":2,"up":150.0,"title":"0.5-150.0","down":0.5}, {"ID":3,"up":300.0,"title":"0.5-300.0","down":0.5}], "arraysize":3} | | | |
| Examples of errors: { "result":-1,"array":[],"arraysize":0} | | | |
| Field name | description | Numeric type | Do you have to |
| desc | description | String | N |
| result | Result information (0 success, failure <0) | Int | Y |
| array | set | JSONArray | Y |
| ID | Quota id, used to return the user quota setting | Int | N |
| title | title | String | N |
| up | Upper limit | Double | N |
| down | Lower limit | Double | N |
| arraysize | Total number | int | Y |

2.7.5 Quota setting interface (/api/quota/set)

| | | | |
|---|--|--------------|----------------|
| Interface name: /api/user/quota/set | | | |
| Example URL: http://api.uobpc.cn/api/quota/set?appid=**&... | | | |
| Input data format: appid ... sign | | | |
| | | | |
| appid | Assign application ID | String | Y |
| username | log-in name | String | Y |
| qids | Quota ID (for multiple groups, separate "1,3,4" by commas) | String | Y |
| sign | Encryption key | String | Y |
| return | | | |
| Successful example: { "result": "0", "desc": "OK" } | | | |
| Examples of errors: { "result": -4, "desc": "signature error!" } | | | |
| | | | |
| Field name | description | Numeric type | Do you have to |
| desc | description | String | Y |
| result | Result information (0 success, failure <0) | Int | Y |

2.8 Single wallet form for merchant users

(implemented by merchant access party)

2.8.1 Message type description

| Code | Description | Gold flow |
|-------------|-------------------------------|----------------------------|
| 1 | Betting | Player deduction |
| 7 | Cancel order (cancel betting) | Player deduction / deposit |
| 9 | Awards | Player deposit |

2.8.2 Merchant user amount query interface (/api/balance)

Access instructions

Merchants must provide this interface when implementing the interface to receive platform balance inquiries.

| | | | |
|---|---|--------|---|
| Interface name: /api/balance is implemented by the merchant company | | | |
| Example URL: http://api.uobpc.cn/api/balance?appid=**&... | | | |
| Input data format: appid ... sign | | | |
| appid | Assign application ID | String | Y |
| username | User name (unique identification) | String | Y |
| notifyid | Message number, the message number of each notification is different, but it must be noted that the re-sending mechanism for network reasons will cause repeated receipt. Merchants can use notifyid to determine to prevent repeated deductions 100032155222301 | String | Y |
| sign | Encryption key | String | Y |

| | | | |
|---|--|--------------|----------------|
| return | | | |
| Successful example: { "result": "0", "desc": "OK", "balance": "20000" } | | | |
| Examples of errors: { "result": -10030, "desc": "Insufficient balance" } | | | |
| Field name | description | Numeric type | Do you have to |
| desc | description | String | Y |
| result | Result information (0 success, failure <0) | Int | Y |
| balance | User balance | double | Y |

2.8.3 Merchant user deduction interface (/api/deduct)

When to use

When a player's account needs to be debited, the platform will use this interface to notify the merchant of the deduction.

At present, when the merchant will be charged:

Player betting (chasing number)

Player rewards

On the notification of deduction, there will not be multiple data for multiple players, but it may be multiple data for one player, because the platform provides a betting area, and players may choose multiple sets of numbers and place bets at once.

Access instructions

Merchants must provide this interface to receive payment notifications from the platform when implementing the interface.

The merchant provides an interface such as (<https://api.uobpc.cn/api/deduct>)

This interface URL is defined and applied by the merchant

The platform sends an HTTP POST packet to the merchant, the data content is POST FORM, and each request can be multiple deduction notifications.

Deduction process description:

Player deduction (betting, tracking number, reward... etc.) confirmation.

The platform side informs the merchant side.

The merchant terminal checks whether the player information (wallet balance... etc.) can bet. If allowed, the deduction will return successfully, and if rejected, the return will fail.

(1) If the deduction is successful, an error code of success (0) will be returned.

(2) When the balance is insufficient, please return the error code of insufficient balance (-1030), and a message of insufficient balance will be displayed to the player.

For other error reasons, please return the error code of the system error (-1001), and the failure message will be displayed to

Players.

According to the received response, the platform implements the corresponding behavior.

| | | | | |
|---|---------------------------------------|--------|---|--|
| Interface name: /api/deduct is implemented by the merchant company | | | | |
| Example URL: | | | | |
| http://api.uobpc.cn/api/deduct?appid=**&... | | | | |
| Input data format: | | | | |
| appid | Assign application ID | String | Y | |
| username | User name (unique identification) | String | Y | |
| amount | Amount (negative number) | Double | Y | |
| notifyid | Message number, the message number of | String | Y | |

| | | | |
|---------------------|--|--------|---|
| | each notification is different, but it must be noted that the re-sending mechanism for network reasons will cause repeated receipt. Merchants can use notifyid to determine to prevent repeated deductions 100032155222301 | | |
| type | Message number | int | Y |
| serialnumber | Requested order number (note order, reward record number) | String | Y |
| sign | Encryption key | String | Y |

| return | | | |
|--|---|--------------|----------------|
| Successful example: | | | |
| { "result": "0", "desc": "OK", "balance": "20000", "orderno": "1000305"} | | | |
| Examples of errors: | | | |
| { "result": -10030, "desc": "Insufficient balance"} | | | |
| Field name | description | Numeric type | Do you have to |
| desc | description | String | Y |
| result | Result information (0 success, failure <0) | Int | Y |
| serialnumber | Merchant's deduction record number (return the merchant's record number for later verification) | String | Y |
| balance | User balance | double | Y |
| orderno | Merchant Record Number | String | Y |

2.8.4 Merchant user deposit interface (/api/deposit)

When to use

When a player's account needs to be deposited, this interface will be used to notify the merchant to deposit. The reason for the deposit may be

1. The award of the player's winning (If the merchant has no betting record in the current period, no notification will be sent, but as long as someone bets in the current period, Even if no one wins the prize, it will still be notified.)
2. Re- award (if the merchant has no betting record in the current period, no notification will be sent).
3. Cancellation of an invalid game (if the merchant has no betting record in this period, no notification will be sent)
4. Before the round is closed, the player takes the initiative to withdraw the order.
5. Refunds for failed deductions on the platform.

Please note that on the deposit notification, there may be multiple data for multiple players, because multiple players may win prizes in the period.

Access instructions

Merchants must provide this interface to receive payment notifications from the platform when implementing the interface.

The merchant provides an interface such as (<https://http://example.com/api/deposit>)

This interface URL is defined and applied by the merchant

The platform sends HTTP POST packets to merchants, the data content is POST FORM, and each request can be multiple

Deposit notice, and there is an additional re-send mechanism to avoid omission of deposit notice.

Deposit process description:

1. The platform side informs the merchant side.
2. The merchant completes the player's deposit and returns a successful (0) error code.
3. The platform side implements the corresponding behavior according to the received response.

If the merchant wants to know the winning amount of the player's odds or the need to store betting details by itself, receive the award

With the deposit notification for the re-awarding, you can check with the betting list:

| |
|---|
| Interface name: /api/deposit is implemented by the merchant company |
| Example URL: http://api.uobpc.cn/api/deposit?appid=**&... |
| Input data format: |

| appid ... sign | | | |
|----------------|--|--------|---|
| appid | Assign application ID | String | Y |
| username | User name (unique identification) | String | Y |
| amount | Amount (positive number) | Double | Y |
| notifyid | Message number, the message number of each notification is different, but it must be noted that the re-sending mechanism for network reasons will cause repeated receipt. Merchants can use notifyid to determine to prevent repeated deductions 100032155222301 | String | Y |
| type | Message number | int | Y |
| serialnumber | Requested order number (note order, reward record number) | String | Y |
| sign | Encryption key | String | Y |

| | | | |
|---|--|--------------|----------------|
| return | | | |
| Successful example: {"result":"0","desc":"OK","balance":"20000","orderno":"1000305"} | | | |
| Examples of errors: {"result":-10001,"desc":"Unknown error"} | | | |
| Field name | description | Numeric type | Do you have to |
| desc | description | String | Y |
| result | Result information (0 success, failure <0) | Int | Y |
| orderno | Merchant record number | String | Y |
| balance | User balance | double | Y |



2.8.5 Merchant user rollback interface (/api/rollback)

When to use

When the payment to the player account fails, this interface will be used to notify the merchant to roll back the amount. Possible reasons for rollback

1. Call the rollback interface when the player requests an abnormal deposit or withdrawal

•Please note that on the notice of rollback, there may be unsuccessful deposits and withdrawals, and you need to verify whether to roll back.

Access instructions

MERCHANTS must provide this interface to receive platform rollback notifications when implementing the interface.

The merchant provides an interface such as
<https://http://example.com/api/rollback>

This interface URL is defined and applied by the merchant

The platform sends HTTP POST packets to merchants, the data content is POST FORM, and each request can be multiple

Deposit notice, and there is an additional re-send mechanism to avoid omission of deposit notice.

Deposit process description:

1. The platform side informs the merchant side.
2. The merchant completes the player's deposit and returns a successful (0) error code.
3. The platform side implements the corresponding behavior according to the received response.

| | | | |
|---|---|--------|---|
| Interface name: /api/rollback is implemented by the merchant company | | | |
| Example URL: http://api.uobpc.cn/api/rollback?appid=**&... | | | |
| Input data format: appid ... sign | | | |
| appid | Assign application ID | String | Y |
| username | User name (unique identification) | String | Y |
| amount | Amount (positive number) | Double | Y |
| notifyid | Message number, the message number of each notification is different, but it must be noted that the re-sending mechanism for network reasons will cause repeated receipt. Merchants can use notifyid to determine to prevent repeated deductions 100032155222301 | String | Y |
| type | Message number | int | Y |
| serialnumber | Requested order number (note order, reward record number) | String | Y |
| errormsg | Error message (URLEncoder) | String | Y |
| sign | Encryption key | String | Y |
| return | | | |

Successful example:

```
{"result":"0","desc":"OK","balance":"20000"}
```

Examples of errors:

```
{"result":-10001,"desc":"Unknown error"}
```

| Field name | description | Numeric type | Do you have to |
|------------|--|--------------|----------------|
| desc | description | String | Y |
| result | Result information (0 success, failure <0) | Int | Y |
| balance | User balance | double | Y |

2.9 Merchant user single wallet JSON format (implemented by merchant access party)

2.9.1 Message type description

| Code | Description | Gold flow |
|------|-------------------------------|----------------------------|
| 1 | Betting | Player deduction |
| 7 | Cancel order (cancel betting) | Player deduction / deposit |
| 9 | Awards | Player deposit |
| 10 | Re-evaluation | Player deposit/deduction |
| 12 | Rollback | Exception rollback |

2.9.2 Merchant user amount query interface (/api/balance)

Access instructions

Merchants must provide this interface when implementing the interface to receive platform balance inquiries.

| | | | |
|---|---|--------|---|
| Interface name: /api/balance is implemented by the merchant company | | | |
| Example URL: http://api.uobpc.cn/api/balance | | | |
| ContentType:application/json | | | |
| Input data format: | | | |
| appid | Assign application ID | String | Y |
| username | User name (unique identification) | String | Y |
| notifyid | Message number, the message number of each notification is different, but it must be noted that the re-sending mechanism for network reasons will cause repeated receipt. Merchants can use notifyid to determine to prevent repeated deductions 100032155222301 | String | Y |
| token | Return the merchant's token | String | N |
| sign | Encryption key | String | Y |

| |
|---|
| return |
| Successful example: { "result": "0", "desc": "OK", "balance": "20000" } |
| Examples of errors: { "result": -10030, "desc": "Insufficient balance" } |
| Field name |
| description |
| Numeric type |
| Do you have to |
| desc |
| description |
| String |
| Y |
| result |
| Result information (0 success, failure <0) |
| Int |
| Y |
| balance |
| User balance |
| double |
| Y |

2.9.3 Merchant user deduction interface (/api/deduct)

When to use

When a player's account needs to be debited, the platform will use this interface to notify the merchant of the deduction.

At present, when the merchant will be charged:

Player betting (chasing number)

Player rewards

On the notification of deduction, there will not be multiple data for multiple players, but it may be multiple data for one player, because the platform provides a betting area, and players may choose multiple sets of numbers and place bets at once.

Access instructions

Merchants must provide this interface to receive payment notifications from the platform when implementing the interface.

The merchant provides an interface such as (<https://api.uobpc.cn/api/deduct>)

This interface URL is defined and applied by the merchant

The platform sends HTTP POST packets to merchants, the data content is POST JSON, and each request can be multiple deduction notifications.

Deduction process description:

Player deduction (betting, tracking number, reward... etc.) confirmation.

The platform side informs the merchant side.

The merchant terminal checks whether the player information (wallet balance... etc.) can bet. If allowed, the deduction will return successfully, and if rejected, the return will fail.

(1) If the deduction is successful, an error code of success (0) will be returned.

(2) When the balance is insufficient, please return the error code of insufficient balance (-1030), and a message of insufficient balance will be displayed to the player.

For other error reasons, please return the error code of the system error (-1001), and the failure message will be displayed to

Players.

According to the received response, the platform implements the corresponding behavior.

| | | | |
|---|-----------------------------------|--------|---|
| Interface name: /api/deduct is implemented by the merchant company | | | |
| Example URL: http://api.uobpc.cn/api/deduct | | | |
| ContentType:application/json | | | |
| Input data format: | | | |
| appid | Assign application ID | String | Y |
| username | User name (unique identification) | String | Y |

| | | | |
|--------------|---|--------|---|
| amount | Amount (negative number) | Double | Y |
| notifyid | Message number, the message number of each notification is different, but it must be noted that the re-sending mechanism for network reasons will cause repeated receipt. Merchants can use notifyid to determine to prevent repeated deductions 100032155222301 | String | Y |
| type | Message number | int | Y |
| serialnumber | Requested order number (note order, reward record number) | String | Y |
| token | Return the merchant's token | String | N |
| sign | Encryption key | String | Y |
| bets | Bet a single JSON object. Not participating in signing . Refer to the betting interface { id bet ID serialnumber bet number userid user ID username username gametype game type gameid game id gamerounno game round number choosebankpoint three public selected villages betpoint betpoint betamount bet amount winlost commamount Commratio occupies a field (the host id can be transmitted based on this (ID10001_0.5)) createtime creation time state order state} | JSON | N |

| | | | |
|--|---|--------------|----------------|
| return | | | |
| Successful example: | | | |
| { "result": "0", "desc": "OK", "balance": "20000", "orderno": "1000305"} | | | |
| Examples of errors: | | | |
| { "result": -10030, "desc": "Insufficient balance"} | | | |
| Field name | description | Numeric type | Do you have to |
| desc | description | String | Y |
| result | Result information (0 success, failure <0>) | Int | Y |
| serialnumber | Merchant's deduction record number (return the merchant's record number for later verification) | String | Y |
| balance | User balance | double | Y |
| orderno | Merchant record number | String | Y |

2.9.4 Merchant user deposit interface (/api/deposit)

When to use

When a player's account needs to be deposited, this interface will be used to notify the merchant to deposit. The reason for the deposit may be

1. The award of the player's winning (If the merchant has no betting record in the current period, no notification will be sent, but as long as someone bets in the current period, Even if no one wins the prize, it will still be notified.)
2. Re- award (if the merchant has no betting record in the current period, no notification will be sent).
3. Cancellation of an invalid game (if the merchant has no betting record in this period, no notification will be sent)
4. Before the round is closed, the player takes the initiative to withdraw the order.
5. Refunds for failed deductions on the platform.

• Please note that on the deposit notification, there may be multiple data for multiple players, because multiple players may win prizes in the period.

Access instructions

Merchants must provide this interface to receive payment notifications from the platform when implementing the interface.

The merchant provides an interface such as (<https://http://example.com/api/deposit>)

This interface URL is defined and applied by the merchant

The platform sends HTTP POST packets to merchants, the data content is POST JSON, and each request can be multiple

Deposit notice, and there is an additional re-send mechanism to avoid omission of deposit notice.

Deposit process description:

1. The platform side informs the merchant side.
2. The merchant completes the player's deposit and returns a successful (0) error code.
3. The platform side implements the corresponding behavior according to the received response.

If the merchant wants to know the winning amount of the player's odds or the need to store betting details by itself, receive the award

With the deposit notification for the re-awarding, you can check with the betting list:

| |
|---|
| Interface name: /api/deposit is implemented by the merchant company |
| Example URL: http://api.uobpc.cn/api/deposit |

| | | | |
|---|--|--------|---|
| ContentType:application/json Input data format: <pre>{"appid":"xtd3G7Z8IUHV","type":"9","amount":"20","serialnumber":"100000123","username":"Z_L_EW","notifyid":"55d73d9add514db8ba9c0bd78035cef2","sign":"b333ad6f5c940d87559baf96a17d6bb2","token":"f333da6f5c940d87559baf96a17d6zz33"}</pre> | | | |
| appid | Assign application ID | String | Y |
| username | User name (unique identification) | String | Y |
| amount | Amount (positive number) | Double | Y |
| notifyid | Message number, the message number of each notification is different, but it must be noted that the re-sending mechanism for network reasons will cause repeated receipt. Merchants can use notifyid to determine to prevent repeated deductions 100032155222301 | String | Y |
| type | Message number | int | Y |
| serialnumber | Requested order number (note order, reward record number) | String | Y |
| token | Return the merchant's token | String | N |
| sign | Encryption key | String | Y |
| bets | Bet a single JSON object. Not participating in signing. Refer to the betting interface { id bet ID serialnumber bet number userid user ID username username gametype game type gameid game id gamerounno game round number choosebankpoint three public selected villages betpoint betpoint betamount bet amount winlost commamount Commratio occupies a field (the host id can be transmitted based on this (ID10001_0.5)) createtime creation time state order state} | JSON | N |

| | | | |
|--|--|--------------|----------------|
| return | | | |
| Successful example: <pre>{"result":"0","desc":"OK","balance":"20000","orderno":"1000305"}</pre> | | | |
| Examples of errors: <pre>{"result":-10030,"desc":"Insufficient balance"}</pre> | | | |
| | | | |
| Field name | description | Numeric type | Do you have to |
| desc | description | String | Y |
| result | Result information (0 success, failure <0) | Int | Y |
| orderno | Merchant record number | String | Y |
| balance | User balance | double | Y |

2.9.5 Merchant user rollback interface (/api/rollback)

When to use

When the payment to the player account fails, this interface will be used to notify the merchant to roll back the amount. Possible reasons for rollback

1. Call the rollback interface when the player requests an abnormal deposit or withdrawal

When the merchant does not respond

When the merchant request timed out

The merchant returns an error

◆Please note that on the notice of rollback, there may be unsuccessful deposits and withdrawals, and you need to verify whether to roll back.

Access instructions

Merchants must provide this interface to receive platform rollback notifications when implementing the interface.

The merchant provides an interface such as (<https://http://example.com/api/rollback>)

This interface URL is defined and applied by the merchant

The platform sends HTTP POST packets to merchants, the data content is POST JSON, and each request can be multiple

Roll back notification, and there is an additional re-send mechanism to avoid omission of deposit notification.

Rollback process description:

1. The platform side informs the merchant side.
2. The merchant will complete the player rollback and return a successful (0) error code.
3. The platform side implements the corresponding behavior according to the received response.

| |
|---|
| Interface name: /api/rollback is implemented by the merchant company |
| Example URL: http://api.uobpc.cn/api/rollback |
| ContentType:application/json Input data format: |

```
{"appid":"xtd3G7Z8IUHV","type":"1","amount":"10","serialnumber":"100000123","username":"Z_L  
EW","notifyid":"55d73d9add514db8ba9c0bd78035cef2","  
errmsg":"fail","sign":"b333ad6f5c940d87559baf96a17d6bb2","token":"f333da6f5c940d87559baf96a17d6  
zz33"}
```

| | | | |
|---------------------|---|--------|---|
| appid | Assign application ID | String | Y |
| username | User name (unique identification) | String | Y |
| amount | Amount (positive number) | Double | Y |
| notifyid | Message number, the message number of each notification is different, but it must be noted that the re-sending mechanism for network reasons will cause repeated receipt. Merchants can use notifyid to determine to prevent repeated deductions 100032155222301 | String | Y |
| type | Message number (follow the last operation type) | int | Y |
| serialnumber | Requested order number (note order, reward record number) | String | Y |
| errmsg | Error message (URLEncoder) | String | Y |
| token | Return the merchant's token | String | N |
| sign | Encryption key | String | Y |

| | | | |
|---|--|---------------------|-----------------------|
| return | | | |
| Successful example: | | | |
| <pre>{"result":"0","desc":"OK","balance":"20000"}</pre> | | | |
| Examples of errors: | | | |
| <pre>{"result":-10001,"desc":"Unknown error"}</pre> | | | |
| Field name | description | Numeric type | Do you have to |
| desc | description | String | Y |
| result | Result information (0 success, failure <0) | Int | Y |
| balance | User balance | double | Y |



2.9.6 Merchant bets active push interface (/api/push/bets)

When to use

When the bet is to be settled, the platform actively requests and carries the json betting information. When the merchant receives the return corresponding value, it will not be pushed repeatedly

| Interface name: /api/push/bets is implemented by the merchant company | | | |
|---|---|--------|---|
| Example URL: http://api.uobpc.cn/api/push/bets | | | |
| ContentType:application/json | | | |
| Input data format: {"appid":"xtd3G7Z8IUHV","notifyid":"55d73d9add514db8ba9c0bd78035cef2","sign":"b333ad6f5c940d87559ba96a17d6bb2","bets":{}} | | | |
| appid | Assign application ID | String | Y |
| notifyid | Message number, the message number of each notification is different, but it must be noted that the re-sending mechanism for network reasons will cause repeated receipt. Merchants can use notifyid to determine to prevent repeated deductions 100032155222301 | String | Y |
| sign | Encryption key | String | Y |
| bets | Bet a single JSON object. Not participating in signing . Refer to the betting interface { id bet ID serialnumber bet number userid user ID username username gametype game type gameid game id gamerounno game round number choosebankpoint three public selected villages betpoint betpoint betamount bet amount winlost commamount Commratio occupies a field (the host id can be transmitted based on this (ID10001_0.5)) createtime creation time state order state} | JSON | N |

| | | | |
|---|-------------|--------------|----------------|
| return | | | |
| Successful example: {"result":"0","desc":"OK"} | | | |
| Examples of errors: {"result":-10001,"desc":"Unknown error"} | | | |
| Field name | description | Numeric type | Do you have to |

| | | | |
|---------------|--|--------|---|
| desc | description | String | Y |
| result | Result information (0 success, failure <0) | Int | Y |

2.10 Get game bet slip interface

(/api/data/getbets)

| Interface name: /data/getbets is implemented by the merchant company | | | |
|--|--|--------------|----------------|
| Example URL: http://api.uobpc.cn/api/data/getbets?prefix=DZ&size=50 | | | |
| Input data format: | | | |
| appid size sign | | | |
| Field name | description | Numeric type | Do you have to |
| appid | Assign application ID | String | Y |
| size | Query size | Int | N |
| sign | Encryption key | String | Y |
| return | | | |
| Web success example: <pre>{"result":0,"array":[{"createtime":"2016-06-28 14:51:53","betpoint":"PLAYER","gamerounid":2098,"commratio":"", "shoenum":6,"shoeid":27,"userid":4,"sbanker":0,"commamount":0.0,"winlost":50.0,"gametype":2,"tabloid":6, choosebankpoint":"","id":12,"serialnumber":12,"state":1,"username":"FAKHD10001","betamount":50.0}],"arraysize":1}</pre> | | | |
| Examples of errors: <pre>{"result":-1,"array":[],"arraysize":0}</pre> | | | |
| Field name | description | Numeric type | Do you have to |
| result | Result information (0 success, -1 failure) | Int | Y |
| arraysize | Collection size | Int | Y |
| array | set | JSONArray | Y |
| id | Betting order ID | Long | N |
| serialnumber | Bet number | Long | N |
| userid | User ID | Long | N |
| username | user name | String | N |
| gametype | game type | Byte | N |
| gameid | Game id | Long | N |
| gamerounno | Game round number | Long | N |
| choosebankpoint | Three public elections | String | N |
| betpoint | Betting point | String | N |
| betamount | Bet amount | Double | N |
| winlost | Win or lose | Double | N |
| commamount | Horse washing amount | Double | N |
| commratio | Occupied as a field (the host id can be transmitted according to this as (ID10001_0.5)) | String | N |

| | | | |
|-------------------|---------------|--|---|
| createtime | Creation time | String | N |
| state | Order Status | -2 to cancel, -1 to discard, 0 to bet, 1 to settle | N |

2.11 Callback processing betting status (/api/data/removebets)

| | | | |
|---|--|--------------|----------------|
| Interface name: /data/removebets Realized by merchant company | | | |
| Example URL: http://api.uobpc.cn/api/data/removebets | | | |
| Input data format: appid ids sign | | | |
| | | | |
| Field name | description | Numeric type | Do you have to |
| appid | Assign application ID | String | Y |
| ids | Consumption bet id (separated by comma) | String | N |
| sign | Encryption key | String | Y |
| return | | | |
| Web success example: { "desc": "ok!", "result": 1 } | | | |
| Examples of errors: { "desc": "error!", "result": -1 } | | | |
| Field name | description | Numeric type | Do you have to |
| desc | description | String | Y |
| result | Result information (0 success, failure <0) | Int | Y |

2.12 Get the game list interface (/api/data/getgames)

| Interface name: /data/getgames is implemented by the merchant company | | | |
|---|---|--------------|----------------|
| Example URL: http://api.uobpc.cn/api/data/getgames?appid=DZ~ifyid=1111110 | | | |
| Input data format: appid notifyid sign | | | |
| Field name | description | Numeric type | Do you have to |
| appid | Assign application ID | String | Y |
| notifyid | Message number, the message number of each notification is different, but it must be noted that the re-sending mechanism for network reasons will cause repeated receipt. Merchants can use notifyid to determine to prevent repeated deductions 100032155222301 | String | Y |
| sign | Encryption key | String | Y |
| return | | | |
| Web success example: {"result":0,"array":[{"id":0,"state":1}],"arraysize":1} | | | |
| Examples of errors: {"result":-1,"array":[],"arraysize":0} | | | |
| Field name | description | Numeric type | Do you have to |
| result | Result information (0 success, -1 failure) | Int | Y |
| arraysize | Collection size | Int | Y |
| array | set | JSONArray | Y |
| id | Betting order ID | Long | N |
| gametype | game type | Byte | N |
| gamestyle | Game style | Byte | N |
| title | Game title | String | N |
| state | Game state (-1/close, 0/idle, 1/open, 2/close, 3/maintenance) | Byte | N |
| betstate | Game betting status(0/off, 1/on) | Byte | N |
| rank | Game sorting (1, 2, 3, 4) | Integer | N |
| updatetime | Change the time | Integer | N |

2.13 Get the game history result interface (/api/record/game/history)

| Interface name: /api/record/game/history Realized by merchant company | | | |
|---|---|--------------|----------------|
| Example URL: http://api.uobpc.cn/api/record/game/history | | | |
| Input data format: appid gameid size sign | | | |
| Field name | description | Numeric type | Do you have to |
| appid | Assign application ID | String | Y |
| notifyid | Message number, the message number of each notification is different, but it must be noted that the re-sending mechanism for network reasons will cause repeated receipt. Merchants can use notifyid to determine to prevent repeated deductions 100032155222301 | String | Y |
| gameid | Game id | Integer | Y |
| index | Current page number (starting from 0) | Integer | Y |
| size | Current query size | Integer | Y |
| sign | Encryption key | String | Y |

| | | | |
|---|--|--------------|----------------|
| return | | | |
| Web success example: { "result":0,"array":[{"gameid":0,"gameresult":"1;2;3;4;5;6;7;8"}],"arraysize":1} | | | |
| Examples of errors: { "result":-1,"array":[],"arraysize":0} | | | |
| Field name | description | Numeric type | Do you have to |
| result | Result information (0 success, -1 failure) | Int | Y |
| arraysize | Collection size | Int | Y |
| array | set | JSONArray | Y |
| gameid | Betting order ID | Long | N |
| gameno | Game period number | Long | N |
| gameresult | Game result | String | N |
| state | Game state(5-finished) | Byte | N |
| finishtime | End Time | Integer | N |

2.14 Get bet statistics interface (/api/record/bets/stat)

| Interface name: /api/record/bets/stat Realized by merchant company | | | |
|---|---|--------------|----------------|
| Example URL: http://api.uobpc.cn/api/record/bets/stat | | | |
| Input data format: appid … sign | | | |
| Field name | description | Numeric type | Do you have to |
| appid | Assign application ID | String | Y |
| notifyid | Message number, the message number of each notification is different, but it must be noted that the re-sending mechanism for network reasons will cause repeated receipt. Merchants can use notifyid to determine to prevent repeated deductions 100032155222301 | String | Y |
| username | log-in name | String | Y |
| begintime | Start time (time stamp, in seconds) | Integer | Y |
| endtime | End time (time stamp, in seconds) | Integer | Y |
| sign | Encryption key | String | Y |

| return | | | |
|---|--|--------------|----------------|
| Web success example: { "result":0,"data":{""betnum":0,"sbetamount":0},"desc":"ok"} | | | |
| Examples of errors: { "result":-1,"data":{},"desc":"error"} | | | |
| Field name | description | Numeric type | Do you have to |
| result | Result information (0 success, -1 failure) | Int | Y |
| desc | description | String | Y |
| data | Data object | JSONObject | Y |
| betnum | Number of bets | Long | N |
| sbetamount | Total amount of bet | Double | N |
| swinlost | Total bet wins and losses | Double | N |

2.15 Get the detailed interface of bets

(/api/record/bets/detail)

| Interface name: /api/record/bets/detail Realized by merchant company | | | |
|---|--|--------------|----------------|
| Example URL: http://api.uobpc.cn/api/record/bets/detail | | | |
| Input data format: appid ... sign | | | |
| | | | |
| Field name | description | Numeric type | Do you have to |
| appid | Assign application ID | String | Y |
| notifyid | Message number, the message number of each notification is different, but it must be noted that the re-sending mechanism for network reasons will cause repeated receipt. Merchants can use notifyid to determine to prevent repeated deductions 100032155222301 | String | Y |
| username | log-in name | String | Y |
| begintime | Start time (time stamp, in seconds) | Integer | Y |
| endtime | End time (time stamp, in seconds) | Integer | Y |
| index | Current page number (starting from 0) | Integer | Y |
| size | Current query size | Integer | Y |
| sign | Encryption key | String | Y |
| return | | | |
| Web success example: {"result":0,"array":[{"gameid":0,"id":1}],"arraysize":1} | | | |
| Examples of errors: {"result":-1,"array":[],"arraysize":0} | | | |
| Field name | description | Numeric type | Do you have to |
| result | Result information (0 success, -1 failure) | Int | Y |
| arraysize | Collection size | Int | Y |
| array | set | JSONArray | Y |

| | | | |
|--------------------|-----------------------------|---------|---|
| id | Bet id | Long | N |
| gameid | Game ID | Long | N |
| gameroundid | Game id | Long | N |
| bettotype | Type of bet | Byte | N |
| betamount | Bet amount | String | N |
| balance | Amount after betting | Double | N |
| betodds | Odds at the time of betting | Double | |
| createtime | Betting time | Integer | |

2.16 Betting follow interface (/api/user/bets/follow)

| | | | |
|---|---|--------------|----------------|
| Interface name: /api/user/bets/follow Realized by merchant company | | | |
| Example URL: http://api.uobpc.cn/api/user/bets/follow | | | |
| Input data format: appid ... sign | | | |
| Field name | description | Numeric type | Do you have to |
| appid | Assign application ID | String | Y |
| notifyid | Message number, the message number of each notification is different, but it must be noted that the re-sending mechanism for network reasons will cause repeated receipt. Merchants can use notifyid to determine to prevent repeated deductions 100032155222301 | String | Y |
| username | log-in name | String | Y |
| betid | Bet id to be followed | Long | Y |
| sign | Encryption key | String | Y |
| return | | | |
| Web success example: { "desc": "ok!", "result": 1 } | | | |
| Examples of errors: { "desc": "error!", "result": -1 } | | | |
| Field name | Field name | Field name | Field name |
| desc | desc | desc | desc |
| result | result | result | result |

2.17 Account details query interface

(/api/record/account/detail)

| Interface name: /api/record/account/detail Realized by merchant company | | | |
|---|---|--------------|----------------|
| Example URL: http://api.uobpc.cn/api/record/account/detail | | | |
| Input data format: appid … sign | | | |
| | | | |
| Field name | description | Numeric type | Do you have to |
| appid | Assign application ID | String | Y |
| notifyid | Message number, the message number of each notification is different, but it must be noted that the re-sending mechanism for network reasons will cause repeated receipt. Merchants can use notifyid to determine to prevent repeated deductions 100032155222301 | String | Y |
| username | log-in name | String | Y |
| begintime | Start time (time stamp, in seconds) | Integer | Y |
| endtime | End time (time stamp, in seconds) | Integer | Y |
| index | Current page number (starting from 0) | Integer | Y |
| size | Current query size | Integer | Y |
| sign | Encryption key | String | Y |
| return | | | |
| Web success example: {"result":0,"array":[{"gameid":0,"id":1}],"arraysize":1} | | | |
| Examples of errors: {"desc":"error!","result":-1} | | | |
| Field name | Field name | Field name | Field name |

| | | | |
|-------------------|--|-----------|---|
| result | Result information (0 success, -1 failure) | Int | Y |
| arraysize | Collection size | Int | Y |
| array | set | JSONArray | Y |
| id | Bill id | Long | N |
| optype | Operation type | Byte | N |
| orderno | Game id | Long | N |
| amount | Change amount | Double | N |
| balance | Amount after change | Double | N |
| createtime | Creation time | Integer | |

2.18 Re-settlement interface (/api/user/bets/settle)

| | | | |
|---|---|--------------|----------------|
| Interface name: /api/user/bets/settle Realized by merchant company | | | |
| Example URL: http://api.uobpc.cn/api/user/bets/settle | | | |
| Input data format: appid … sign | | | |
| Field name | description | Numeric type | Do you have to |
| appid | Assign application ID | String | Y |
| notifyid | Message number, the message number of each notification is different, but it must be noted that the re-sending mechanism for network reasons will cause repeated receipt. Merchants can use notifyid to determine to prevent repeated deductions 100032155222301 | String | Y |
| id | Betting order ID | Long | Y |
| sign | Encryption key | String | Y |

| | | | |
|---|---------------|---------------|---------------|
| return | | | |
| Web success example: { "desc": "ok!", "result": 1 } | | | |
| Examples of errors: { "desc": "error!", "result": -1 } | | | |
| | | | |
| Field name | Field name | Field name | Field name |
| desc | desc | desc | desc |
| result | result | result | result |



3. Appendix reference

3.1 Error code

| Code | Applicable interface | description |
|-------|----------------------|--|
| 0 | All interfaces | success |
| -1000 | All interfaces | Unknown system abnormality, please contact |
| -1001 | All interfaces | System internal abnormality, please contact |
| -1002 | All interfaces | Parameter error, please check the passed parameter |
| -1005 | All interfaces | Appid error |
| -1006 | All interfaces | Appid does not exist, please contact |
| -1007 | All interfaces | Signature error, please check the signature |
| -1008 | All interfaces | Expired |
| -1009 | All interfaces | Request too frequent |
| -1010 | All interfaces | Illegal IP |
| -1011 | User interface | Not registered, please register first |
| -1012 | User interface | User does not exist |
| -1013 | User interface | User already exists |
| -1014 | User interface | User banned |
| -1015 | User interface | User password error |
| -1020 | Wallet interface | Insufficient user balance |
| -1021 | Wallet interface | The deposit and withdrawal amount is non-zero |
| -1022 | User interface | Limit does not exist |
| -1030 | Wallet interface | Insufficient deduction amount |
| -1040 | Verification code | Verification code error |
| -1041 | All interfaces | Illegal domain name |

3.2 Language code

| Value | Language description |
|-------|----------------------|
| 1 | Simplified Chinese |
| 2 | English |
| 3 | Korean |
| 4 | Japanese |
| 5 | Thai |
| | |

3.3 Game Type

| Value | Game Description |
|-------|------------------|
| 4 | 百家乐 |
| 5 | 轮盘 |
| 10 | 龙虎 |
| 11 | 骰宝 |
| 14 | 牛牛 |
| 17 | 三公 |

3.4 Game Result Code

| Value | Diamonds♦ | Clubs♣ | Hearts♥ | Spades♠ |
|-------|-----------|--------|---------|---------|
| A | 1 | 2 | 3 | 4 |
| 2 | 5 | 6 | 7 | 8 |
| 3 | 9 | 10 | 11 | 12 |
| 4 | 13 | 14 | 15 | 16 |
| 5 | 17 | 18 | 19 | 20 |
| 6 | 21 | 22 | 23 | 24 |
| 7 | 25 | 26 | 27 | 28 |
| 8 | 29 | 30 | 31 | 32 |
| 9 | 33 | 34 | 35 | 36 |
| 10 | 37 | 38 | 39 | 40 |
| J | 41 | 42 | 43 | 44 |
| Q | 45 | 46 | 47 | 48 |
| K | 49 | 50 | 51 | 52 |

3.5 Account operation type

| Type of | Type description |
|--------------|--------------------------|
| 1 | deposit |
| 2 | Withdrawal |
| 3 | Betting |
| 4 | Settlement |
| 5 | Cancel result |
| 6 | Corrected result |
| 7 | Abnormal rollback amount |
| 8 | Cancel a bet |
| 9 | Amount transfer |
| 20 | Accounted for |
| twenty one | Agent rebate |
| twenty two | Agent return rebate |
| twenty three | Deliver benefits |
| twenty four | Send back water |
| 25 | Payout |
| 28 | Personal tips |
| | |

3.6 Currency code

| sequence | Currency code | 中文描述 | English description |
|----------|---------------|---------|-----------------------------|
| 1 | AED | 阿联酋迪拉姆 | United Arab Emirates Dirham |
| 2 | AFN | 阿富汗尼 | Afghan Afghani |
| 3 | ALL | 阿尔巴尼列克 | Albania Lek |
| 4 | AMD | 亚美尼亚德拉姆 | Armenia Dram |
| 5 | ANG | 荷兰盾 | Dutch Guilder |
| 6 | AOA | 安哥拉宽扎 | Angola Kwanza |
| 7 | ARS | 阿根廷比索 | Argentina Peso |
| 8 | AUD | 澳元 | Australia Dollar |
| 9 | AWG | 阿鲁巴弗罗林 | Aruba Florin |
| 10 | AZN | 阿塞拜疆马纳特 | Azerbaijan Manat |
| 11 | BAM | 波黑可兑换马克 | Bosnia Convertible Mark |
| 12 | BBD | 巴巴多斯元 | Barbados Dollar |
| 13 | BDT | 孟加拉国塔卡 | Bangladesh Taka |
| 14 | BGN | 保加利亚列弗 | Bulgaria Lev |
| 15 | BHD | 巴林第纳尔 | Bahrain Dinar |
| 16 | BIF | 布隆迪法郎 | Burundi Franc |
| 17 | BMD | 百慕大元 | Bermudian Dollar |

| | | | |
|----|-----|----------|---------------------------|
| 18 | BND | 文莱元 | Brunei Dollar |
| 19 | BOB | 玻利维亚诺 | Bolivian Boliviano |
| 20 | BRL | 巴西雷亚尔 | Brazilian Real |
| 21 | BSD | 巴哈马元 | Bahamian Dollar |
| 22 | BTN | 不丹努扎姆 | Bhutanese Ngultrum |
| 23 | BWP | 博茨瓦纳普拉 | Botswana Pula |
| 24 | BYR | 白俄罗斯卢布 | Belarusian Ruble |
| 25 | BZD | 伯利兹元 | Belize Dollar |
| 26 | CAD | 加元 | Canadian Dollar |
| 27 | CDF | 刚果法郎 | Congolese Franc |
| 28 | CHF | 瑞士法郎 | Swiss Franc |
| 29 | CLF | 智利比索(基金) | Chilean Unidad de Fomento |
| 30 | CLP | 智利比索 | Chilean Peso |
| 31 | CNH | 中国离岸人民币 | Chinese Offshore Renminbi |
| 32 | CNY | 人民币 | Chinese Yuan |
| 33 | COP | 哥伦比亚比索 | Colombia Peso |
| 34 | CRC | 哥斯达黎加科朗 | Costa Rica Colon |
| 35 | CUP | 古巴比索 | Cuban Peso |
| 36 | CVE | 佛得角埃斯库多 | Cape Verde Escudo |
| 37 | CYP | 塞普路斯镑 | Cyprus Pound |
| 38 | CZK | 捷克克朗 | Czech Republic Koruna |
| 39 | DEM | 德国马克 | Deutsche Mark |
| 40 | DJF | 吉布提法郎 | Djiboutian Franc |
| 41 | DKK | 丹麦克朗 | Danish Krone |
| 42 | DOP | 多米尼加比索 | Dominican Peso |
| 43 | DZD | 阿尔及利亚第纳尔 | Algerian Dinar |
| 44 | ECS | 厄瓜多尔苏克雷 | Ecuadorian Sucre |
| 45 | EGP | 埃及镑 | Egyptian Pound |
| 46 | ERN | 厄立特里亚纳克法 | Eritrean Nakfa |
| 47 | ETB | 埃塞俄比亚比尔 | Ethiopian Birr |
| 48 | EUR | 欧元 | Euro |
| 49 | FJD | 斐济元 | Fiji Dollar |
| 50 | FKP | 福克兰群岛镑 | Falkland Islands Pound |
| 51 | FRF | 法国法郎 | French Franc |
| 52 | GBP | 英镑 | British Pound |
| 53 | GEL | 格鲁吉亚拉里 | Georgian Lari |
| 54 | GHS | 加纳塞地 | Ghanaian Cedi |
| 55 | GIP | 直布罗陀镑 | Gibraltar Pound |
| 56 | GMD | 冈比亚达拉西 | Gambian Dalasi |
| 57 | GNF | 几内亚法郎 | Guinean Franc |
| 58 | GTQ | 危地马拉格查尔 | Guatemalan Quetzal |
| 59 | GYD | 圭亚那元 | Guyanese Dollar |
| 60 | HKD | 港币 | Hong Kong Dollar |
| 61 | HNL | 洪都拉斯伦皮拉 | Honduran Lempira |
| 62 | HRK | 克罗地亚库纳 | Croatian Kuna |
| 63 | HTG | 海地古德 | Haitian Gourde |
| 64 | HUF | 匈牙利福林 | Hungarian Forint |
| 65 | IDR | 印度尼西亚卢比 | Indonesian Rupiah |
| 66 | IEP | 爱尔兰镑 | Irish Pound |
| 67 | ILS | 以色列新谢克尔 | Israeli New Shekel |
| 68 | INR | 印度卢比 | Indian Rupee |
| 69 | IQD | 伊拉克第纳尔 | Iraqi Dinar |
| 70 | IRR | 伊朗里亚尔 | Iranian Rial |
| 71 | ISK | 冰岛克郎 | Icelandic Krona |

| | | | |
|-----|-----|-----------|------------------------------------|
| 72 | ITL | 意大利里拉 | Italian Lira |
| 73 | JMD | 牙买加元 | Jamaican Dollar |
| 74 | JOD | 约旦第纳尔 | Jordanian Dinar |
| 75 | JPY | 日元 | Japanese Yen |
| 76 | KES | 肯尼亚先令 | Kenyan Shilling |
| 77 | KGS | 吉尔吉斯斯坦索姆 | Kyrgyzstani Som |
| 78 | KHR | 柬埔寨瑞尔 | Cambodian Riel |
| 79 | KMF | 科摩罗法郎 | Comorian franc |
| 80 | KPW | 朝鲜元 | North Korean Won |
| 81 | KRW | 韩元 | South Korean Won |
| 82 | KWD | 科威特第纳尔 | Kuwaiti Dinar |
| 83 | KYD | 开曼群岛元 | Cayman Islands Dollar |
| 84 | KZT | 哈萨克斯坦坚戈 | Kazakstani Tenge |
| 85 | LAK | 老挝基普 | Lao kip |
| 86 | LBP | 黎巴嫩镑 | Lebanese Pound |
| 87 | LKR | 斯里兰卡卢比 | Sri Lankan Rupee |
| 88 | LRD | 利比里亚元 | Liberian dollar |
| 89 | LSL | 莱索托洛蒂 | Lesotho Loti |
| 90 | LTL | 立陶宛立特 | Lithuanian Litas |
| 91 | LVL | 拉脱维亚拉特 | Latvian Lats |
| 92 | LYD | 利比亚第纳尔 | Libyan Dinar |
| 93 | MAD | 摩洛哥迪拉姆 | Moroccan Dirham |
| 94 | MDL | 摩尔多瓦列伊 | Moldovan Leu |
| 95 | MGA | 马达加斯加阿里亚里 | Malagasy Ariary |
| 96 | MKD | 马其顿代纳尔 | Macedonian Denar |
| 97 | MMK | 缅甸元 | Myanmar Kyat |
| 98 | MNT | 蒙古图格里克 | Mongolian Tugrik |
| 99 | MOP | 澳门元 | Macau Pataca |
| 100 | MRO | 毛里塔尼亚乌吉亚 | Mauritania Ouguiya |
| 101 | MUR | 毛里求斯卢比 | Mauritian Rupee |
| 102 | MVR | 马尔代夫拉菲亚 | Maldives Rufiyaa |
| 103 | MWK | 马拉维克瓦查 | Malawian Kwacha |
| 104 | MXN | 墨西哥比索 | Mexican Peso |
| 105 | MXV | 墨西哥(资金) | Mexican Unidad De Inversion |
| 106 | MYR | 林吉特 | Malaysian Ringgit |
| 107 | MZN | 莫桑比克新梅蒂卡尔 | New Mozambican Metical |
| 108 | NAD | 纳米比亚元 | Namibian Dollar |
| 109 | NGN | 尼日利亚奈拉 | Nigerian Naira |
| 110 | NIO | 尼加拉瓜新科多巴 | Nicaraguan Cordoba Oro |
| 111 | NOK | 挪威克朗 | Norwegian Krone |
| 112 | NPR | 尼泊尔卢比 | Nepalese Rupee |
| 113 | NZD | 新西兰元 | New Zealand Dollar |
| 114 | OMR | 阿曼里亚尔 | Omani Rial |
| 115 | PAB | 巴拿马巴波亚 | Panamanian Balboa |
| 116 | PEN | 秘鲁新索尔 | Peruvian Nuevo Sol |
| 117 | PGK | 巴布亚新几内亚基那 | Papua New Guinea Kina |
| 118 | PHP | 菲律宾比索 | Philippine Peso |
| 119 | PKR | 巴基斯坦卢比 | Pakistan Rupee |
| 120 | PLN | 波兰兹罗提 | Polish Zloty |
| 121 | PYG | 巴拉圭瓜拉尼 | Paraguayan Guarani |
| 122 | QAR | 卡塔尔里亚尔 | Qatari Riyal |
| 123 | RON | 罗马尼亚列伊 | Romanian Leu |
| 124 | RSD | 塞尔维亚第纳尔 | Serbian Dinar |
| 125 | RUB | 俄罗斯卢布 | Russian Ruble |

| | | | |
|-----|-----|----------|-----------------------------------|
| 126 | RWF | 卢旺达法郎 | Rwandan Franc |
| 127 | SAR | 沙特里亚尔 | Saudi Arabian Riyal |
| 128 | SBD | 所罗门群岛元 | Solomon Islands Dollar |
| 129 | SCR | 塞舌尔卢比 | Seychelles Rupee |
| 130 | SDG | 苏丹磅 | Sudanese Pound |
| 131 | SEK | 瑞典克朗 | Swedish Krona |
| 132 | SGD | 新加坡元 | Singapore Dollar |
| 133 | SHP | 圣赫勒拿镑 | Saint Helena Pound |
| 134 | SIT | 斯洛文尼亚托拉尔 | Slovenian Tolar |
| 135 | SLL | 塞拉利昂利昂 | Sierra Leonean Leone |
| 136 | SOS | 索马里先令 | Somali Shilling |
| 137 | SRD | 苏里南元 | Suriname Dollar |
| 138 | STD | 圣多美多布拉 | Sao Tome Dobra |
| 139 | SVC | 萨尔瓦多科朗 | Salvadoran Colon |
| 140 | SYP | 叙利亚镑 | Syrian Pound |
| 141 | SZL | 斯威士兰里兰吉尼 | Swazi Lilangeni |
| 142 | THB | 泰铢 | Thai Baht |
| 143 | TJS | 塔吉克斯坦索莫尼 | Tajikistan Somoni |
| 144 | TMT | 土库曼斯坦马纳特 | Turkmenistan Manat |
| 145 | TND | 突尼斯第纳尔 | Tunisian Dinar |
| 146 | TOP | 汤加潘加 | Tongan Pa'Anga |
| 147 | TRY | 土耳其里拉 | Turkish Lira |
| 148 | TTD | 特立尼达多巴哥元 | Trinidad and Tobago Dollar |
| 149 | TWD | 新台币 | New Taiwan Dollar |
| 150 | TZS | 坦桑尼亚先令 | Tanzanian Shilling |
| 151 | UAH | 乌克兰格里夫纳 | Ukrainian Hryvnia |
| 152 | UGX | 乌干达先令 | Ugandan Shilling |
| 153 | USD | 美元 | United States Dollar |
| 154 | UYU | 乌拉圭比索 | Uruguayan Peso |
| 155 | UZS | 乌兹别克斯坦苏姆 | Uzbekistani Som |
| 156 | VEF | 委内瑞拉玻利瓦尔 | Venezuelan Bolivar Fuerte |
| 157 | VND | 越南盾 | Viet Nam Dong |
| 158 | VUV | 瓦努阿图瓦图 | Vanuatu Vatu |
| 159 | WST | 萨摩亚塔拉 | Samoa Tala |
| 160 | XAF | 中非法郎 | Central African CFA Franc |
| 161 | XAG | 银价盎司 | Ounces of Silver |
| 162 | XAU | 金价盎司 | Ounces of Gold |
| 163 | XCD | 东加勒比元 | East Caribbean Dollar |
| 164 | XCP | 铜价盎司 | Ounces of Copper |
| 165 | XDR | IMF特别提款权 | IMF Special Drawing Rights |
| 166 | XOF | 西非法郎 | West African CFA |
| 167 | XPD | 钯价盎司 | Ounces of Palladium |
| 168 | XPF | 太平洋法郎 | French Pacific Franc |
| 169 | XPT | 铂价盎司 | Ounces of Platinum |
| 170 | YER | 也门里亚尔 | Yemeni Rial |
| 171 | ZAR | 南非兰特 | South African Rand |
| 172 | ZMW | 赞比亚克瓦查 | Zambian Kwacha |
| 173 | ZWL | 津巴布韦元 | Zimbabwean Dollar |