

NG interface phpdemo

Note: This is an example of a PHP-based NGAPI interface encapsulation that provides common operational encapsulations for NG gaming APIs, including:

Create a member account (register)
Query balance (getbalance)
Query the balance of the entire platform (getbalanceall)
Get the game login address (gamelogin)
Query real-time betting records (gamerealtimerecords)
Get the game code (gamecode)
Query the merchant platform quota

Code structure: The NewNgApi class encapsulates all API-related functions. API URL configuration is defined in the constructor, including the addresses of each interface. sendRequest method handles API requests uniformly, uses cURL to send POST requests, and the data format is JSON. Sample call: Directly instantiate NewNgApi, and then call gamelogin("username") to get the game login address. The result is parsed and output in json_decode(\$res, true). How to use it?

```
<?php
$api = new NewNgApi();
$res = $api->gamelogin("a123456");
print_r(json_decode($res, true));
```

This code will obtain the game login address of the player a123456 and return the JSON format response (note that before obtaining the player's address, it will be determined internally whether the player is registered, whether the player needs to convert the amount and other logical operations).

NewNgApi.php

```
<?php
date_default_timezone_set('PRC');

/**
 * NGAPI PHP Demo class.
 *
 * @link https://wiki.neapi.com
 * @package NewNgApi
 * @version (PHP 7+, PHP 8+)
 * @author https://wiki.neapi.com
 * @copyright https://www.neapi.com
 * @since 1.0.0
 */
class NewNgApi
{
    protected $creatememberaccountUrl; // Create a member account
    protected $getbalanceUrl; // Query the balance
    protected $getplatformbalanceUrl; // Query the balance of the entire
    platform
```

```
protected $conversionallUrl; // One-click transfer of the full platform
quota
protected $gameloginUrl; // Get the game login address
protected $demologinUrl; // Get the demo game login address
protected $gameorderUrl; // Game order query
protected $gamerealtimerecordsUrl; // Real-time game record
protected $gamehistoryrecordsUrl; // Historical game records
protected $conversionUrl; // quota conversion
protected $conversionStatusUrl; // Quota conversion status query
protected $gamecodeUrl; // Get the game code
protected $getmerchantBalanceUrl; // Check the merchant balance

protected $apiUrl; // API interface domain name
protected $sn; // The merchant prefix after registration is activated,
please log in to the management background to view
protected $secretKey; // After registration and activation, please log
in to the management background to view
protected $platType; // Set platform type All lowercase
protected $currency; // Currency code BRL: Brazilian real, CNY: RMB,
HKD: Hong Kong dollar, IDR: Indonesian rupiah, INR: Indian rupee, MMK:
Burmese kyat, NGN: Nigerian naira, PHP: Philippine peso, THB: Thai
baht, USD: US dollar, VND: Vietnamese dong
protected $sign; // 32-bit lowercase md5(random+sn+secretKey)
protected $random; // Random string 16-32 bits lowercase letters +
numbers

    public function __construct()
    {
$this->apiUrl = 'interface domain name'; // API interface domain name
$this->sn = 'Merchant Prefix'; // Merchant Prefix after registration
and activation, please log in to the management background to view
$this->secretKey = 'Merchant Key'; // After registration and
activation, please log in to the management background to view
$this->currency = "CNY"; // 货币
$this->platType = 'ag'; // Game parameters in
"https://wiki.neapi.com/doku.php?id=Game Platform"
        $this->random =
substr(str_shuffle('abcdefghijklmnopqrstuvwxyz0123456789'), 0, rand(16,
32));
        $this->sign = md5($this->random . $this->sn .
$this->secretKey);
        $this->creatememberaccountUrl = $this->apiUrl .
"/api/server/create";
        $this->getbalanceUrl = $this->apiUrl . "/api/server/balance";
        $this->getplatformbalanceUrl = $this->apiUrl .
'/api/server/balanceAll';
        $this->conversionallUrl = $this->apiUrl .
'/api/server/transferAll';
        $this->gameloginUrl = $this->apiUrl . '/api/server/gameUrl';
        $this->demologinUrl = $this->apiUrl . '/api/server/demoUrl';
        $this->gameorderUrl = $this->apiUrl .
```

```

'/api/server/recordOrder';
    $this->gamerealtimerecordsUrl = $this->apiUrl .
'/api/server/recordAll';
    $this->gamehistoryrecordsUrl = $this->apiUrl .
'/api/server/recordHistory';
    $this->conversionUrl = $this->apiUrl . '/api/server/all-
credit';
    $this->conversionStatusUrl = $this->apiUrl .
'/api/server/record';
    $this->gamecodeUrl = $this->apiUrl . '/api/server/gameCode';
    $this->getmerchantBalanceUrl = $this->apiUrl .
'/api/server/quota';
}

protected $game_type_live = '1';           //视讯
protected $game_type_slot = '2';           //Slot machine
protected $game_type_lottery = '3';         //彩票
protected $game_type_sports = '4';          //体育
protected $game_type_esports = '5';         //电竞
protected $game_type_fishing = '6';         //捕鱼
protected $game_type_poker = '7';          //棋牌

/**
 * Create a player
 * Type Field Name Required Description
 * @param string $platType is a game platform, refer to the appendix of
"Game Platform"
 * @param string $playerId is Player account, length limit 5-11 digits
Lowercase letters + numeric combination
 * @param string $currency is game currency, refer to the appendix of
"Game Platform"
 * mixed
 */
public function register($username)
{
    $data = array(
        "playerId" => $username,
        "platType" => $this->platType,
        "currency" => $this->currency
    );
    $res = $this->sendRequest($this->creatememberaccountUrl,
$data);
    return $res;
}

/**
 * Check balance
 * Type Field Name Required Description
 * @param platType string is a game platform (refer to the appendix of
"Game Platform")
 * @param playerId string is player account

```

```
*@param currency string Yes Currency code BRL: Brazilian real, CNY:
RMB, HKD: Hong Kong dollar, IDR: Indonesian rupiah, INR: Indian rupee,
MMK: Burmese kyat, NGN: Nigerian naira, PHP: Philippine peso, THB: Thai
baht, USD: US dollar, VND: Vietnamese dong
*/
public function getbalance($username)
{
    $data = array(
        "platType" => $this->platType,
        "playerId" => $username,
        "currency" => $this->currency,
    );
    $res = $this->sendRequest($this->getbalanceUrl, $data);
    return $res;
}

/**
* Check the balance of the entire platform
* Type Field Name Required Description
*@param playerId string is player account
*@param currency string Yes Currency code BRL: Brazilian real, CNY:
RMB, HKD: Hong Kong dollar, IDR: Indonesian rupiah, INR: Indian rupee,
MMK: Burmese kyat, NGN: Nigerian naira, PHP: Philippine peso, THB: Thai
baht, USD: US dollar, VND: Vietnamese dong
*/
public function getbalanceall($username)
{
    $data = array(
        "playerId" => $username,
        "currency" => $this->currency,
    );
    $res = $this->sendRequest($this->getbalanceUrl, $data);
    return $res;
}

/**
*Get the game login address
*Type Field Name Required Description
*@param string $platType game platform (refer to the appendix of "Game
Platform") (required)
*@param string $playerId Player Account (required)
*@param string $gameType game type, (1: video, 2: slot machine, 3:
lottery, 4: sports, 5: e-sports, 6: fishing, 7: chess and cards)
(required)
*@param string $lang game language, default simplified Chinese (refer
to the appendix of "Game Language")
*@param string $gameCode game code, default game hall (refer to the
appendix of "Game Platform")
*@param string $returnUrl The address of the game exit is jumped,
example: "https://www.neapi.com"
*@param string $ingress terminal type, device1: computer web version,
```

```

device2: mobile web version (for other specific terminals, please refer
to the appendix of the "Game Platform") (required)
*/
@param string $oddsType Lottery handicap, A: (default), B, C, only IG
lottery and SGWin lottery are available
@return mixed
*/
public function gamelogin($username, $lang = "", $gameCode = "",
$returnUrl = "", $ingress = 'device1', $oddsType = "")
{
    $data = array(
        "platType" => $this->platType,
        "playerId" => $username,
        "gameType" => $this->game_type_live,
        "currency" => $this->currency,
        "lang" => $lang,
        "gameCode" => $gameCode,
        "returnUrl" => $returnUrl,
        "ingress" => $ingress,
        "oddsType" => $oddsType
    );
    $res = $this->sendRequest($this->gameloginUrl, $data);
    return $res;
}

/**
 * Get real-time betting history
 * Type Field Name Required Description
 * @param $currency
 * @param $pageNo
 * @param int $pageSize
 * @return mixed
 */
public function gamerealtimerecords($page = 1, $pageSize = 200)
{
    $data = array(
        "currency" => $this->currency,
        "pageNo" => $page,
        "pageSize" => $pageSize,
    );
    $res = $this->sendRequest($this->gamerealtimerecordsUrl,
$data);
    return $res;
}

/**
 *Get the game code
 *@param string $platType game platform, refer to the appendix of "Game
Platform"
 *@return mixed
 */

```

```
public function gamecode()
{
    $data = array(
        "platType" => $this->platType
    );
    $res = $this->sendRequest($this->gamecodeUrl, $data);
    return $res;
}

/**
 * @return mixed
 * Check the merchant platform quota
 */
public function quota()
{
    $data = array();
    $res = $this->sendRequest($this->getmerchantBalanceUrl, $data);
    return $res;
}

private function sendRequest($url, $data = array())
{
    // Request header
    $header = [
        'sn:' . $this->sn, // If authentication is required
        'sign:' . $this->sign,
        'random:' . $this->random,
        'Content-Type: application/json', //Request type
    ];
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 60);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, FALSE);
    curl_setopt($ch, CURLOPT_HTTPHEADER, $header);
    curl_setopt($ch, CURLOPT_POSTFIELDS, empty($data) ? "{}" :
    json_encode($data));
    $contents = curl_exec($ch);
    curl_close($ch);
    return $contents;
}

/**
 * Use demo
 */
// $api = new NewNgApi();
// $res = $api->register("username");
```

```
$api = new NewNgApi();  
$res = $api->gamelogin("username");  
echo $res; //Return json by default (refer to the document API to  
return content)  
  
//print_r(json_decode($res, true)); //Return in array form
```

From:

<https://doc.neapi.com/> - **NG游戏接口API文档**

Permanent link:

<https://doc.neapi.com/doku.php?id=en:demov2>

Last update: **2025/02/08 01:21**

