

PHP对接腾讯云短信实现短信发送

注：代码是根据我的需求进行编程的，所以要根据各自需求进行更改。

在开始编程之前首先你应该具备以下条件：

- 腾讯云账号



- 腾讯云平台申请创建签名（需审核）



- 腾讯云平台申请创建短信模板（需审核）



- 领取一个免费的短信包（根据情况，我是先行测试，有条件也可以直接付费买。不是必要条件）



开发步骤：

1. 下载腾讯发送短信的封装类文件

作者的实现的基本要求是：

- 国内短信
- 指定模板单发短信

所以就只用了两个文件：

SmsSingleSender.php



根据实际使用框架，做相应改动

```
<?php
```

```
namespace Admin\Controller\YDX;
```

```
use Admin\Controller\YDX\SmsSenderUtil; //务必要引用SmsSenderUtil
```

```

/**
 * 单发短信类
 *
 */
class SmsSingleSender
{
    private $url;
    private $appid;
    private $appkey;
    private $util;

    /**
     * 构造函数
     *
     * @param string $appid sdkappid
     * @param string $appkey sdkappid对应的appkey
     */
    public function __construct($appid, $appkey)
    {
        $this->url = "https://yun.tim.qq.com/v5/tlssmssvr/sendsms";
        $this->appid = $appid;
        $this->appkey = $appkey;
        $this->util = new SmsSenderUtil();
    }

    /**
     * 普通单发
     *
     * 普通单发需明确指定内容，如果有多个签名，请在内容中以 【】 的方式添加到信息内容
     中，否则系统将使用默认签名。
     *
     * @param int $type 短信类型，0 为普通短信，1 营销短信
     * @param string $nationCode 国家码，如 86 为中国

```

* @param string \$phoneNumber 不带国家码的手机号

* @param string \$msg 信息内容，必须与申请的模板格式一致，否则将返回错误

* @param string \$extend 扩展码，可填空串

* @param string \$ext 服务端原样返回的参数，可填空串

* @return string 应答json字符串，详细内容参见腾讯云协议文档

*/

```
public function send($type, $nationCode, $phoneNumber, $msg, $extend = "", $ext = "")
```

```
{
```

```
    $random = $this->util->getRandom();
```

```
    $curTime = time();
```

```
    $wholeUrl = $this->url . "?sdkappid=" . $this->appid . "&random=" . $random;
```

```
    // 按照协议组织 post 包体
```

```
    $data = new \stdClass();
```

```
    $tel = new \stdClass();
```

```
    $tel->nationcode = "".$nationCode;
```

```
    $tel->mobile = "".$phoneNumber;
```

```
    $data->tel = $tel;
```

```
    $data->type = (int)$type;
```

```
    $data->msg = $msg;
```

```
    $data->sig = hash("sha256",
```

```
        "appkey=".$this->appkey."&random=".$random."&time="
```

```
        ".$curTime."&mobile=".$phoneNumber, FALSE);
```

```
    $data->time = $curTime;
```

```
    $data->extend = $extend;
```

```
    $data->ext = $ext;
```

```
    return $this->util->sendCurlPost($wholeUrl, $data);
```

```
}
```

```
/**
```

* 指定模板单发

*

* @param string \$nationCode 国家码, 如 86 为中国

* @param string \$phoneNumber 不带国家码的手机号

* @param int \$templId 模板 id

* @param array \$params 模板参数列表, 如模板 {1}...{2}...{3}, 那么需要带三个参数

* @param string \$sign 签名, 如果填空串, 系统会使用默认签名

* @param string \$extend 扩展码, 可填空串

* @param string \$ext 服务端原样返回的参数, 可填空串

* @return string 应答json字符串, 详细内容参见腾讯云协议文档

*/

```
public function sendWithParam($nationCode, $phoneNumber, $templId = 0,
$params,
```

```
    $sign = "", $extend = "", $ext = "")
```

```
{
```

```
    $random = $this->util->getRandom();
```

```
    $curTime = time();
```

```
    $wholeUrl = $this->url . "?sdkappid=" . $this->appid . "&random=" . $random;
```

```
// 按照协议组织 post 包体
```

```
$data = new \stdClass();
```

```
$tel = new \stdClass();
```

```
$tel->nationcode = "". $nationCode;
```

```
$tel->mobile = "". $phoneNumber;
```

```
$data->tel = $tel;
```

```
$data->sig = $this->util->calculateSigForTempl($this->appkey, $random,
    $curTime, $phoneNumber);
```

```
$data->tpl_id = $templId;
```

```
$data->params = $params;
```

```
$data->sign = $sign;
```


```
$data->time = $curTime;
```

```
$data->extend = $extend;
```

```
$data->ext = $ext;

return $this->util->sendCurlPost($wholeUrl, $data);
}
}
```

SmsSenderUtil.php

```
 <?php
namespace Admin\Controller\YDX;

/**
 * 发送Util类
 *
 */
class SmsSenderUtil
{
    /**
     * 生成随机数
     *
     * @return int 随机数结果
     */
    public function getRandom()
    {
        return rand(100000, 999999);
    }

    /**
     * 生成签名
     *
     * @param string $appkey    sdkappid对应的appkey
     * @param string $random    随机正整数
     * @param string $curTime   当前时间
     */
}
```

**@param array \$phoneNumbers 手机号码*

**@return string 签名结果*

**/*

public function calculateSig(\$appkey, \$random, \$curTime, \$phoneNumbers)

{

 \$phoneNumbersString = \$phoneNumbers[0];

 for (\$i = 1; \$i < count(\$phoneNumbers); \$i++) {

 \$phoneNumbersString .= ("," . \$phoneNumbers[\$i]);

 }

 return hash("sha256", "appkey=" . \$appkey . "&random=" . \$random

 ."&time=" . \$curTime . "&mobile=" . \$phoneNumbersString);

}

*/***

** 生成签名*

**@param string \$appkey sdkappid对应的appkey*

**@param string \$random 随机正整数*

**@param string \$curTime 当前时间*

**@param array \$phoneNumbers 手机号码*

**@return string 签名结果*

**/*

public function calculateSigForTemplAndPhoneNumbers(\$appkey, \$random,

 \$curTime, \$phoneNumbers)

{

 \$phoneNumbersString = \$phoneNumbers[0];

 for (\$i = 1; \$i < count(\$phoneNumbers); \$i++) {

 \$phoneNumbersString .= ("," . \$phoneNumbers[\$i]);

 }

 return hash("sha256", "appkey=" . \$appkey . "&random=" . \$random

 ."&time=" . \$curTime . "&mobile=" . \$phoneNumbersString);

```
}
```

```
public function phoneNumbersToArray($nationCode, $phoneNumbers)
```

```
{
```

```
    $i = 0;
```

```
    $tel = array();
```

```
    do {
```

```
        $telElement = new \stdClass();
```

```
        $telElement->nationcode = $nationCode;
```

```
        $telElement->mobile = $phoneNumbers[$i];
```

```
        array_push($tel, $telElement);
```

```
    } while (++$i < count($phoneNumbers));
```

```
    return $tel;
```

```
}
```

```
/**
```

```
 * 生成签名
```

```
 *
```

```
 * @param string $appkey    sdkappid对应的appkey
```

```
 * @param string $random    随机正整数
```

```
 * @param string $curTime    当前时间
```

```
 * @param array $phoneNumber 手机号码
```

```
 * @return string 签名结果
```

```
 */
```

```
public function calculateSigForTempl($appkey, $random, $curTime,  
$phoneNumber)
```

```
{
```

```
    $phoneNumbers = array($phoneNumber);
```

```
    return $this->calculateSigForTemplAndPhoneNumbers($appkey, $random,  
        $curTime, $phoneNumbers);
```

```
}
```



```

/**
 * 生成签名
 *
 * @param string $appkey    sdkappid对应的appkey
 * @param string $random    随机正整数
 * @param string $curTime   当前时间
 * @return string 签名结果
 */
public function calculateSigForPuller($appkey, $random, $curTime)
{
    return hash("sha256", "appkey=".$appkey."&random=".$random
        . "&time=".$curTime);
}

```

```

/**
 * 生成上传文件授权
 *
 * @param string $appkey    sdkappid对应的appkey
 * @param string $random    随机正整数
 * @param string $curTime   当前时间
 * @param array $fileSha1Sum 文件sha1sum
 * @return string 授权结果
 */
public function calculateAuth($appkey, $random, $curTime, $fileSha1Sum)
{
    return hash("sha256", "appkey=".$appkey."&random=".$random
        . "&time=".$curTime."&content-sha1=".$fileSha1Sum);
}

```

```

/**
 * 生成sha1sum
 *

```

```

* @param string $content 内容
* @return string 内容sha1散列值
*/
public function sha1sum($content)
{
    return hash("sha1", $content);
}

/**
* 发送请求
*
* @param string $url 请求地址
* @param array $dataObj 请求内容
* @return string 应答json字符串
*/
public function sendCurlPost($url, $dataObj)
{
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $url);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_CONNECTTIMEOUT, 60);
    curl_setopt($curl, CURLOPT_POSTFIELDS, json_encode($dataObj));
    curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 0);

    $ret = curl_exec($curl);
    if (false == $ret) {
        // curl_exec failed
        $result = "{ \"result\": \"-2\", \"errmsg\": \"\" . curl_error($curl) . \"\"}";
    } else {

```

```

    $rsp = curl_getinfo($curl, CURLINFO_HTTP_CODE);
    if (200 != $rsp) {
        $result = "{ \"result\": \"-1\", \"errmsg\": \"\". $rsp
            . \"\" . curl_error($curl) . \"\"}";
    } else {
        $result = $ret;
    }
}

curl_close($curl);

return $result;
}

/**
 * 发送请求
 *
 * @param string $req 请求对象
 * @return string 应答json字符串
 */
public function fetch($req)
{
    $curl = curl_init();

    curl_setopt($curl, CURLOPT_URL, $req->url);
    curl_setopt($curl, CURLOPT_HTTPHEADER, $req->headers);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $req->body);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_CONNECTTIMEOUT, 60);
    curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 0);

```

```

$result = curl_exec($curl);

if (false == $result) {
    // curl_exec failed
    $result = "{ \"result\": \"-2\", \"errmsg\": \"\" . curl_error($curl) . \"\"}";
} else {
    $code = curl_getinfo($curl, CURLINFO_HTTP_CODE);
    if (200 != $code) {
        $result = "{ \"result\": \"-1\", \"errmsg\": \"\" . $code
            . \"\" . curl_error($curl) . \"\"}";
    }
}

curl_close($curl);

return $result;
}
}

```

2. 写一个公共的用来调用该文件



根据实际使用框架，做相应改动

```

<?php
namespace Admin\Controller\YDX;

use Admin\Controller\YDX\SmsSingleSender;

class TenCentYunDuanXin
{
    // 短信应用SDK AppID
    private $appid = 1400007;

    // 短信应用SDK AppKey

```

```
private $appkey = "67b1eec*****890f0d74";
```

// 短信模板ID，需要在短信应用中申请。NOTE: 这里的模板ID`7839`只是一个示例，真实的模板ID需要在短信控制台中申请

```
private $templateId = 1137727;
```

// 签名内容 NOTE: 这里的签名只是示例，请使用真实的已申请的签名，签名参数使用的是`签名内容`，而不是`签名ID`

```
private $smsSign = "****sksks";
```

/* TODO 腾讯云短信API(国内)--使用指定模板ID单发短信

* 如果出现短信发送错误的情况,请查看AppID、AppKey、\$templateId、\$smsSign是否有误

* 参数列:

* \$phoneNumber 手机号码

* \$params 短信内容中需要插入的参数，按顺序放入该一维数组中

* 例如：您的验证码为{1}，请于{2}分钟内输入。\$params=['验证码' , '有效时间']

*/

```
public function Send($phoneNumber,array $params){
```

```
// 指定模板ID单发短信
```

```
$sender = new SmsSingleSender($this->appid, $this->appkey);
```

```
$result = $sender->sendWithParam("86", $phoneNumber, $this->templateId,
```

```
$params, $this->smsSign, "", ""); // 签名参数未提供或者为空时，会使用默认签名发送短信
```

```
$rsp = json_decode($result,true);
```

```
if($rsp['result']==0)
```

```
    return true;
```

```
else
```

```
    return $rsp['errmsg'];
```

```
}
```

```
}
```

然后调用封装好的 *TenCentYunDuanXin* 接口就好了！不要忘了传递参数呦！