互亿无线身份证四要素认证 API

接口说明

|  |  |
| --- | --- |
| 文件标识 | Ihuyi\_IDCARD API |
| 版本编号 | V1.2 |
| 发布者 | 上海思锐信息技术有限公司 |
| 发布日期 | 2023-05-021 |
| 最新更新 | 2025-05-26 |

目录

[1.概要 3](#_Toc6274)

[1.1文档说明 3](#_Toc30250)

[1.2提交方式 3](#_Toc8309)

[1.3加密方式 3](#_Toc13822)

[1.4 API ID/ API KEY 3](#_Toc23957)

[2.接交认证 3](#_Toc32237)

[2.1接口描述 3](#_Toc10359)

[2.2请求地址 3](#_Toc21332)

[2.3请求参数 4](#_Toc11197)

[2.4响应示例 5](#_Toc20025)

[2.5响应参数 5](#_Toc32657)

[2.6 GET请求示例 6](#_Toc3409)

[3.余额查询 6](#_Toc31274)

[3.1接口描述 6](#_Toc16135)

[3.2请求地址 6](#_Toc9510)

[3.3请求参数 6](#_Toc4058)

[3.4响应示例 7](#_Toc1787)

[3.5响应参数 7](#_Toc4238)

[3.6 GET请求示例 7](#_Toc10461)

[4.生成动态密码方式 7](#_Toc31494)

[5.如何实现UNIX时间戳 8](#_Toc17597)

[6.DEMO示例 9](#_Toc12986)

[7.aes加密示例 10](#_Toc9291)

[7.1 PHP示例 10](#_Toc1583)

[7.2 java示例 10](#_Toc16686)

[8.联系方式 10](#_Toc8410)

# 1.概要

## 1.1文档说明

本文档主要提供给互亿平台的用户对接接口的使用说明，开发者可以利用互亿无线提供的 HTTP 接口，调用互亿无线的实名认证服务。

## 1.2提交方式

GET / POST

## 1.3加密方式

1、采用HTTPS协议提交请求

2、通过MD5动态签名方式加密

## 1.4 API ID/ API KEY

登录用户中心，进入【实名认证】模块，在【产品总览】页面右侧，如下图所示：



# 2.接交认证

## 2.1接口描述

通过POST、 GET 方式进行提交，数据编码采用utf-8格式，24小时全天提交。

内容格式为：Content-Type: application/x-www-form-urlencoded

## 2.2请求地址

https://api.ihuyi.com/idcard/Submit.json

## 2.3请求参数

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数值描述** | **必填** |
| account | APIID（用户中心【实名认证】-【产品总览】查看） | 是 |
| password | 1、APIKEY（用户中心【实名认证】-【产品总览】查看）  2、动态密码（生成动态密码方式请看该文档末尾的说明） | 是 |
| app | id4 | 是 |
| name | 身份证姓名 | 是 |
| id\_card\_no | 身份证号码 | 是 |
| begin\_date | 有效起始日期，格式：20050712 | 是 |
| end\_date | 有效截止日期，格式：20050712 (长期有效身份证填 8 个 0) | 是 |
| time | Unix时间戳（10位整型数字，当使用动态密码方式时为必填） | 否 |

用户通过http 或 https提交请求到服务器后，服务器会返回一串json格式的字符串，SubmitResult结构说明如下：

|  |  |  |
| --- | --- | --- |
| **参数名称** | **类型** | **描述** |
| code | int | 返回值为2时，表示提交成功 |
| idcardid | string | 当提交成功后，此字段为流水号，否则为0 |
| msg | string | 提交结果描述 |
| result | object | 验证结果 |
| result.status | int | 状态码  2验证一致  1验证不一致 |
| result.remark | string | 描述 |

## 2.4响应示例

示例（成功）

{"code":2,"msg":"提交成功","idcardid":"16236437872836","result":{"status":2,"remark":"验证一致":}}

示例（失败）

{"code":405,"msg":"用户名或密码不正确","idcardid":"0","result":{"status":0,"remark":""}}

## 2.5响应参数

|  |  |
| --- | --- |
| **code** | **msg** |
| 0 | 未知错误 |
| 1 | 提交失败 |
| 2 | 提交成功 |
| 400 | 非法ip访问 |
| 401 | 帐号不能为空 |
| 402 | 密码不能为空 |
| 403 | 应用不存在 |
| 404 | 姓名不能为空 |
| 4041 | 姓名格式不正确（2-10个中文汉字） |
| 406 | 身份证号码不能为空 |
| 4061 | 身份证号码格式不正确 |
| 4070 | 有效起始日期不能为空 |
| 4071 | 有效截止日期不能为空 |
| 4072 | 有效起始日期格式不正确 |
| 4073 | 有效截止日期格式不正确 |
| 408 | 查询无结果 |
| 405 | API ID或API KEY不正确 |
| 4050 | 账号被冻结 |
| 40501 | 动态密码已过期 |
| 40502 | 动态密码校验失败 |
| 40505 | 没有签定合同 |
| 4051 | 剩余条数不足 |
| 4052 | 访问ip与备案ip不符 |
| 4086 | 通道无效 |
| 4010 | 通道错误 |

## 2.6 GET请求示例

https://api.ihuyi.com/idcard/Submit.json?account=APIID&password=APIKEY&app=id4&name=姓名&id\_card\_no=401123343333xxxxxxxxxxxxxx&begin\_date=20200202&end\_date=20230909

# 3.余额查询

## 3.1接口描述

通过接口可以方便的查询你的账户剩余条数。

内容格式为：Content-Type: application/x-www-form-urlencoded

## 3.2请求地址

https://api.ihuyi.com/idcard/GetNum.json

## 3.3请求参数

|  |  |  |
| --- | --- | --- |
| **参数名称** | **参数值描述** | **必填** |
| account | APIID（用户中心【实名认证】-【产品总览】查看） | 是 |
| password | 1、APIKEY（用户中心【实名认证】-【产品总览】查看）  2、动态密码（生成动态密码方式请看该文档末尾的说明） | 是 |
| app | id4 | 是 |
| time | Unix时间戳（10位整型数字，当使用动态密码方式时为必填） | 否 |

用户通过http 或 https提交请求到服务器后，服务器会返回一串json格式的字符串，GetNumResult结构说明如下：

|  |  |  |
| --- | --- | --- |
| **参数名称** | **类型** | **描述** |
| code | int | 状态返回值 |
| msg | string | 查询结果描述 |
| num | string | 剩余数量 |

## 3.4响应示例

示例（成功）

{"code":2,"msg":"查询成功","num":123}

示例（失败）

{"code":405,"msg":"用户名或密码不正确","num":0}

## 3.5响应参数

|  |  |
| --- | --- |
| **code** | **msg** |
| 2 | 查询成功 |
| 400 | 非法ip访问 |
| 401 | 帐号不能为空 |
| 402 | 密码不能为空 |
| 403 | 应用不存在 |
| 405 | API ID或API KEY不正确 |
| 4050 | 账号被冻结 |
| 40501 | 动态密码已过期 |
| 40502 | 动态密码校验失败 |
| 4052 | 访问ip与备案ip不符 |

## 3.6 GET请求示例

https://api.ihuyi.com/idcard/GetNum.json?account=APIID&password=APIKEY&app=id4

# 4.生成动态密码方式

以PHP为例，提交认证的生成方式：

$time=time();

$password=md5($account.$apikey.$app.$name.$id\_card\_no.$time);

示例：

APIID： test

APIKEY：1q784322ba1\*\*\*\*\*\*

app：id4

name：张三

id\_card\_no：40122111\*\*\*\*\*\*

time：1651544941

结果：$password=md5('test1q784322ba1\*\*\*\*\*\*id4张三40122111\*\*\*\*\*\*1651544941');

注意：所以字符编码统一为UTF-8

查询剩余条数的生成方式：

$time=time();

$password=md5($account.$apikey.$app.$time);

示例：

APIID： test

APIKEY：1q784322ba1\*\*\*\*\*\*

app：id4

time：1651544941

结果：$password=md5('test1q784322ba1\*\*\*\*\*\*id41651544941');

Unix时间戳参考地址：<http://tool.chinaz.com/Tools/unixtime.aspx>

# 5.如何实现UNIX时间戳

|  |  |
| --- | --- |
| **语言** | **代码** |
| JAVA | String date = new java.text.SimpleDateFormat("dd/MM/yyyy HH:mm:ss").format(new java.util.Date(Unix timestamp \*1000)) |
| Linux | date -d @Unix timestamp |
| C | strptime(datetime, "%Y-%m-%d %H:%M:%S", &tm\_time);  unixtime = mktime(&tm\_time); |
| C++ | std::time\_t t = std::time(0); |
| DELPHI | uses DateUtils;  DateTimeToUnix(Now) |
| GO | import "time"  cur := time.Now()  timestamp := cur.UnixNano() / 1000000 |
| Perl | 先 my $time = Unix timestamp 然后  my ($sec, $min, $hour, $day, $month, $year) =  (localtime($time))[0,1,2,3,4,5,6] |
| PHP | date('r', Unix timestamp) |
| Node.js | var date = new Date( time \* 1000 );  var time = date.getTime()/1000; |
| Python | 先 import time 然后 time.gmtime(Unix timestamp) |
| Ruby | Time.at(Unix timestamp) |
| SHELL | date +%s //UNIX的时间戳; |
| VBScript / ASP | DateAdd("s", Unix timestamp, "01/01/1970 00:00:00") |
| VB6 | DateDiff("s", "1970-1-1 0:0:0", ToUnixTime) |
| 易语言 | 时间\_到时间戳（） |

# 6.DEMO示例

|  |  |
| --- | --- |
| **语言** | **DEMO地址** |
| PHP | https://www.ihuyi.com/demo/idcard4/php.html |
| JAVA | https://www.ihuyi.com/demo/idcard4/java.html |
| C++ | https://www.ihuyi.com/demo/idcard4/cpp.html |
| GO | https://www.ihuyi.com/demo/idcard4/go.html |
| python | https://www.ihuyi.com/demo/idcard4/python.html |

# 7.aes加密示例

## 7.1 PHP示例

<?php

function aes\_encode($text){

$key = 'APIKEY';

$cipher = "aes-256-cbc";

$options = OPENSSL\_RAW\_DATA;

$iv = substr(md5($key), 8, 16);

$encrypted = openssl\_encrypt($text, $cipher, $key, $options, $iv);

return base64\_encode($encrypted);

}

$name = aes\_encode(‘张三’);

$id\_card\_no = aes\_encode(‘40122111\*\*\*\*\*\*1651544941’);

?>

## 7.2 java示例

请访问“身份证四要素认证接口文档”文件夹下的“DEMO -> java”查看。

# 8.联系方式

上海思锐信息技术有限公司  
销售热线： 4008-808-898  
技术支持： 021-5169 8929  
网站网址：<http://www.ihuyi.com/>