

API 产品使用指南

(身份证二要素实名校验)

目录

| | |
|------------------------|----|
| 一、文档概述..... | 3 |
| 二、产品购买..... | 3 |
| 三、获取授权密钥 APPCODE | 4 |
| 四、获取 API 文档 | 4 |
| 1. 身份证二要素校验..... | 4 |
| 五、调用 API | 5 |
| 1. 示例代码 Curl 版 | 5 |
| 2. 示例代码 Java 版..... | 5 |
| 3. 示例代码 C#版..... | 6 |
| 4. 示例代码 PHP 版 | 8 |
| 5. 示例代码 Python 版 | 8 |
| 6. 示例代码 ObjectC 版..... | 9 |
| 六、常见问题..... | 10 |
| 1. 售后和服务 | 10 |
| 2. 产品续费 | 10 |
| 3. 使用情况查询..... | 11 |
| 4. 发票申请 | 11 |
| 5. 可用次数扣减规则..... | 11 |
| 附录一 | 12 |
| 1. 响应状态码定义..... | 12 |

一、文档概述

本使用指南将快速引导您如何调用使用华辰网络在阿里云云市场开放的 API 接口服务。您需要依次完成以下步骤：



调用 API 接口的四个基本要素：

- 调用地址
- 请求方式
- 请求参数
- 授权密钥

以下会详细介绍如何获取四个基本要素，并提供 API 调用 Demo 供参考。

二、产品购买

从阿里云云市场购买 API 服务，在浏览器访问产品页面，选择合适套餐并点击立即购买按钮，提交订单完成支付即可完成购买流程。

身份证二要素 (元素) 实名认证校验检测

身份证二要素 (元素) 实名认证校验检测，核验姓名、身份证信息是否匹配一致。实时联网公安数据核查，零存储，毫秒级响应速度，保证真实，保证产品准确率。

¥ **1600**

用户评分：★★★★★
近180天成交：951笔

套餐版本：
免费试用 2元 / 10次 180元 / 1000次 850元 / 5000次 **1600元 / 10000次**
3000元 / 20000次 7000元 / 50000次 13000元 / 100000次 24000元 / 200000次
55000元 / 500000次 更多套餐请联系客服

套餐配额：**10000次**

购买时长：**单次** 自购买之日起 1年有效

立即购买 加入购物车

产品页面：

<https://market.aliyun.com/products/57000002/cmapi025567.html>

三、获取授权密钥 APPCODE

授权，是指授予您调用某个 API 接口的权限。您购买了阿里云云市场的 API 服务就意味着您已经获得了授权。

购买成功之后您进入云市场的管理控制台，就可以看见您购买的所有 API 服务。您可以在当前控制台查看到相关的授权密钥 APPCODE。



管理控制台：<https://market.console.aliyun.com>

四、获取 API 文档

1. 身份证二要素校验

调用地址：<https://api12.aliyun.venusn.com/cert/id-card>

请求方式：GET

返回类型：JSON

请求参数 (Header):

| 名称 | 类型 | 是否必须 | 描述 |
|---------------|--------|------|---|
| Authorization | String | True | 调用授权身份认证，值为 “APPCODE + 半角空格 + APPCODE 值” |

请求参数 (Query):

| 名称 | 类型 | 是否必须 | 描述 |
|--------|--------|------|-------|
| name | String | True | 姓名 |
| number | String | True | 身份证号码 |

响应参数:

| | 名称 | 类型 | 描述 |
|---|-----|--------|-----------------|
| ○ | ret | Number | 响应状态码，见附录一状态码定义 |
| ○ | msg | String | 响应状态描述 |

| | | | |
|---|--------|--------|----------------------------|
| ○ | log_id | String | 交易流水号 |
| ○ | data | Object | 响应数据 |
| - | code | String | 校验结果编号 (0:一致 1:不一致 3:校验失败) |
| - | desc | String | 校验结果描述 |

五、调用 API

通过上述步骤，您已经获取了授权密钥 APPCODE、API 定义文档。

您可以直接使用产品页面上为您提供多语言调用示例代码进行调用；也可以自行编辑 HTTP 请求来调用 API。

目前提供了 Curl、Java、C#、PHP、Python、ObjectC 共 6 种语言示例代码。

使用示例代码进行调用，注意将示例代码中的 APPCODE 密钥参数值替换为步骤三中获取的授权密钥 APPCODE。

1. 示例代码 Curl 版

```
1. curl -i -k --get --include 'https://api12.aliyun.venuscn.com/cert/id-card?name=name&number=number' -H 'Authorization:APPCODE 你自己的AppCode'
```

2. 示例代码 Java 版

```
1. public static void main(String[] args) {
2.     String host = "https://api12.aliyun.venuscn.com";
3.     String path = "/cert/id-card";
4.     String method = "GET";
5.     String appcode = "你自己的AppCode";
6.     Map<String, String> headers = new HashMap<String, String>();
7.     //最后在 header 中的格式(中间是英文空格)为
8.     Authorization:APPCODE 83359fd73fe94948385f570e3c139105
9.     headers.put("Authorization", "APPCODE " + appcode);
10.    Map<String, String> querys = new HashMap<String, String>();
11.    querys.put("name", "name");
12.    querys.put("number", "number");
13.
14.    try {
```

```

15.     /**
16.     * 重要提示如下:
17.     * HttpUtils 请从
18.     * https://github.com/aliyun/api-gateway-demo-sign-
    java/blob/master/src/main/java/com/aliyun/api/gateway/demo/util/HttpUtils.java
19.     * 下载
20.     *
21.     * 相应的依赖请参照
22.     * https://github.com/aliyun/api-gateway-demo-sign-
    java/blob/master/pom.xml
23.     */
24.     HttpResponse response = HttpUtils.doGet(host, path, method, headers,
    querys);
25.     System.out.println(response.toString());
26.     //获取 response 的 body
27.     //System.out.println(EntityUtils.toString(response.getEntity()));
28. } catch (Exception e) {
29.     e.printStackTrace();
30. }
31. }

```

3. 示例代码 C#版

```

1. //using System.IO;
2. //using System.Text;
3. //using System.Net;
4. //using System.Net.Security;
5. //using System.Security.Cryptography.X509Certificates;
6.
7.     private const String host = "https://api12.aliyun.venuscn.com";
8.     private const String path = "/cert/id-card";
9.     private const String method = "GET";
10.    private const String appcode = "你自己的 AppCode";
11.
12.    static void Main(string[] args)
13.    {
14.        String querys = "name=name&number=number";
15.        String bodys = "";
16.        String url = host + path;
17.        HttpWebRequest httpRequest = null;
18.        HttpResponseMessage httpResponse = null;
19.
20.        if (0 < querys.Length)

```

```
21.         {
22.             url = url + "?" + querys;
23.         }
24.
25.         if (host.Contains("https://"))
26.         {
27.             ServicePointManager.ServerCertificateValidationCallback = ne
w RemoteCertificateValidationCallback(CheckValidationResult);
28.             httpRequest = (HttpWebRequest)WebRequest.CreateDefault(new U
ri(url));
29.         }
30.         else
31.         {
32.             httpRequest = (HttpWebRequest)WebRequest.Create(url);
33.         }
34.         httpRequest.Method = method;
35.         httpRequest.Headers.Add("Authorization", "APPCODE " + appcode);
36.
37.         if (0 < bodys.Length)
38.         {
39.             byte[] data = Encoding.UTF8.GetBytes(bodys);
40.             using (Stream stream = httpRequest.GetRequestStream())
41.             {
42.                 stream.Write(data, 0, data.Length);
43.             }
44.         }
45.         try
46.         {
47.             httpResponse = (HttpWebResponse)httpRequest.GetResponse();
48.         }
49.         catch (WebException ex)
50.         {
51.             httpResponse = (HttpWebResponse)ex.Response;
52.         }
53.         Console.WriteLine(httpResponse.StatusCode);
54.         Console.WriteLine(httpResponse.Method);
55.         Console.WriteLine(httpResponse.Headers);
56.         Stream st = httpResponse.GetResponseStream();
57.         StreamReader reader = new StreamReader(st, Encoding.GetEncoding(
"utf-8"));
58.         Console.WriteLine(reader.ReadToEnd());
59.         Console.WriteLine("\n");
60.
```

```

61.     }
62.
63.     public static bool CheckValidationResult(object sender, X509Certificate certificate, X509Chain chain, SslPolicyErrors errors)
64.     {
65.         return true;
66.     }

```

4. 示例代码 PHP 版

```

1. <?php
2.     $host = "https://api12.aliyun.venuscn.com";
3.     $path = "/cert/id-card";
4.     $method = "GET";
5.     $appcode = "你自己的 AppCode";
6.     $headers = array();
7.     array_push($headers, "Authorization:APPCODE " . $appcode);
8.     $querys = "name=name&number=number";
9.     $bodys = "";
10.    $url = $host . $path . "?" . $querys;
11.
12.    $curl = curl_init();
13.    curl_setopt($curl, CURLOPT_CUSTOMREQUEST, $method);
14.    curl_setopt($curl, CURLOPT_URL, $url);
15.    curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);
16.    curl_setopt($curl, CURLOPT_FAILONERROR, false);
17.    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
18.    curl_setopt($curl, CURLOPT_HEADER, true);
19.    if (1 == strpos("$".$host, "https://"))
20.    {
21.        curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
22.        curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, false);
23.    }
24.    var_dump(curl_exec($curl));
25. ?>

```

5. 示例代码 Python 版

```

1. import urllib, urllib2, sys
2. import ssl
3.
4.
5. host = 'https://api12.aliyun.venuscn.com'
6. path = '/cert/id-card'

```



```

7. method = 'GET'
8. appcode = '你自己的 AppCode'
9. querys = 'name=name&number=number'
10. bodys = {}
11. url = host + path + '?' + querys
12.
13. request = urllib2.Request(url)
14. request.add_header('Authorization', 'APPCODE ' + appcode)
15. ctx = ssl.create_default_context()
16. ctx.check_hostname = False
17. ctx.verify_mode = ssl.CERT_NONE
18. response = urllib2.urlopen(request, context=ctx)
19. content = response.read()
20. if (content):
21.     print(content)

```

6. 示例代码 ObjectC 版

```

1. NSString *appcode = @"你自己的 AppCode";
2. NSString *host = @"https://api12.aliyun.venuscn.com";
3. NSString *path = @"/cert/id-card";
4. NSString *method = @"GET";
5. NSString *querys = @"?name=name&number=number";
6. NSString *url = [NSString stringWithFormat:@"%s%s%s", host, path, querys]
;
7. NSString *bodys = @"";
8.
9. NSMutableURLRequest *request = [NSMutableURLRequest requestWithURL:[NSURL UR
LWithString: url] cachePolicy:1 timeoutInterval: 5];
10. request.HTTPMethod = method;
11. [request addValue: [NSString stringWithFormat:@"APPCODE %@", appcode] f
orHTTPHeaderField: @"Authorization"];
12. NSURLSession *requestSession = [NSURLSession sessionWithConfiguration:[NSURL
SessionConfiguration defaultSessionConfiguration]];
13. NSURLSessionDataTask *task = [requestSession dataTaskWithRequest:request
14. completionHandler:^(NSData * _Nullable body, NSURLResponse * _Nullable
response, NSError * _Nullable error) {
15.     NSLog(@"Response object: %@", response);
16.     NSString *bodyString = [[NSString alloc] initWithData:body encoding:NSUTF
8StringEncoding];
17.
18.     //打印应答中的 body
19.     NSLog(@"Response body: %@", bodyString);
20. }];

```

21.

22. [task resume];

六、常见问题

1. 售后和服务

在产品使用过程中，如遇到任何问题，均可通过店铺首页或产品页面的在线旺旺客服、客服电话、电子邮箱等方式与我们联系。

店铺首页：<https://shop49910m19.market.aliyun.com>

产品页面：

<https://market.aliyun.com/products/57000002/cmapi025567.html>

客服电话：0755-86665320、4009030002 转 13145

电子邮箱：support@venuscn.com

2. 产品续费

在产品服务可使用次数不足 20%的情况下，阿里云账户所绑定的手机号码及电子邮箱均会收到余量不足预警通知。

如需续费，可进入管理控制台找到需要续费的产品服务，点击右上角购买套餐包进入到产品页面（或者在浏览器访问产品页面），选择合适套餐并点击立即购买按钮，提交订单完成支付即可完成续费流程；生产成功后可使用次数会自动累加。

管理控制台：<https://market.console.aliyun.com>

产品页面：

<https://market.aliyun.com/products/57000002/cmapi025567.html>

3. 使用情况查询

如需查询产品服务的使用情况，可进入管理控制台找到对应的产品服务，可用看到该产品服务状态、到期时间、使用情况（统计会有延迟情况）以及购买套餐包列表等信息。



管理控制台：<https://market.console.aliyun.com>

4. 发票申请

如需申请开具发票，可进入到阿里云控制台-云市场发票索取页面，选择服务商-深圳华辰网络科技有限公司，勾选需要开具发票的订单，点击下一步填写开票信息并提交即可。

如申请开具发票金额 ≤ 200 元，将默认发顺丰到付快递。

增值税专用发票、增值税普通发票（金额 ≥ 1000 元）统一为纸质发票；
增值税普通发票（金额 < 1000 元），默认为增值税电子普通发票。

接收到开票申请后，正常情况下财务统一会在每周五开具并寄出（节假日推迟）；如有紧急情况，可以通过售后和服务的联系方式与客服联系并确认。

云市场发票索取页面：

<https://usercenter2.aliyun.com/invoice/create-by-detail/yun-market>

5. 可用次数扣减规则

阿里云的次数扣减是根据 HTTP 响应状态码来判断的，只有当 HTTP 响应状态码为 $[200, 300)$ 时才会扣减次数，所以调用时需处理 HTTP 响应状态码为 200 外的响应，获取响应 JSON 数据。

HTTP 响应状态码错误代码表参考：

https://help.aliyun.com/document_detail/43906.html

附录一

1. 响应状态码定义

| 状态码 | 描述信息 |
|-------|-------------------|
| 200 | 处理成功 |
| 40001 | 姓名不能为空 |
| 40002 | 姓名格式错误 |
| 40003 | 身份证号码不能为空 |
| 40004 | 身份证号码格式错误 |
| 40020 | 其他错误（具体以响应参数错误为准） |
| 40099 | 校验认证失败 |