

码号状态极速检测 API 接口规范

北京云网极速科技有限公司

联系电话：010-53630311

目录

1	空号检测 API 接口发送.....	错误！未定义书签。
1.1	空号检测 API 接口提交地址.....	3
1.2	参数定义.....	3
1.3	空号检测 API 接口提交响应.....	3
1.3.1	格式说明.....	错误！未定义书签。
1.3.2	示例.....	4
1.3.3	响应状态值说明.....	5
1.4	注意事项.....	5
1.5	空号检测 API 接口发送例子：	5

本文档主要提供空号检测 API 和定义说明。

1 空号检测 API 接口请求

用户在遵循 HTTPS 协议的前提下，可通过 POST 方式提交空号检测 API 请求。

1.1 空号检测 API 接口提交地址

URL 地址为：

<https://kh.ywjisu.com/Api/ApiAuth/getCheckResult> (安全)

1.2 参数定义

序号	参数	说明
1	code	必填参数。用户 API 账号
2	pwd	必填参数。用户 API 密码
3	mobiles	必填参数。合法的手机号码（支持一次传输多个手机号码逗号分隔，多个手机号码英文逗号分隔，一次最多 50 个手机号码）
4	timestamp	必填时间戳 10 位，
6	token	必填公钥，md5(API 账号+私钥+时间戳)

1.3 空号检测 API 接口提交响应

空号检测 API 通过 https 请求提交到服务器后，服务器返回响应码，响应码的格式如下：

响应消息, 响应状态, 响应结果

resultMsg, resultCode , resultObj

1.3.1 示例

➤ 响应结果 JSON 实例

```
{
  "resultMsg": "成功",
  "resultCode": "000000",
  "resultObj": [
    {
      "mobile": "13817367247",
      "lastTime": 1509361607000,
      "area": "上海-上海",
      "numberType": "中国移动 GSM",
      "chargesStatus": "1",
      "status": "1"
    }
  ]
}
```

结果状态	说明
chargesStatus	1 : 收费 0: 不收费
status	0: 空号 1: 实号 2: 停机 3: 库无 4: 沉默号 5: 风险号

空号即为近一周内出现过空号的号码。

实号即为活跃号码（命中准确率 100%，误差率小于 5%），出现误差原因：可能意外停机。

停机即为近一周内出现过停机的号码（命中准确率 100%，误差率小于 5%）。出现误差的原因，可能停机的手机号恢复使用。

库无是指利用 API 接口检测中，系统无法在数据库中查询到该号码（可能号码输入错误等原因），如手机号检测反馈状态为库无，不收取检测费用。

沉默号里面分为两种状态，一种是超过半年没有激活过的空号，二是近三月流量均消耗低于 30M，且近三个月平均流量低于 30M 的用户也有可能随时停机，简单的来说就是快投入市场的空号、小号、新号、非智能机客户，我们认为这种是对移动互联网没有太大的兴趣人群。不建议高价值的推广宣传。

风险号属于长时间关机或未开通语音服务以及易投诉的用户。

1.3.2 响应状态值说明

代码	说明
200000	请求成功
400001	参数校验异常，请检查参数
400101	商户信息不存在
400102	商户 IP 地址信息不合法
400200	黑名单列表
400201	手机号码不能为空
400901	账户信息不存在
400902	应用类型信息不存在
403000	用户校验失败，请联系客服
403101	账户被系统禁用
403102	账户未激活
415000	请求数据转换异常
500000	系统异常
500002	数据处理异常
500003	业务操作失败
500004	远程调用失败
500005	账户余额异常
500006	请求外部系统失败
500901	邮箱未设置
500902	账户信息已存在
500903	账户相关能力已激活
504000	系统超时
510101	用户产品可用数量不足

1.4 注意事项

如果运营商返回状态延时，则需要等待 3 秒重新请求，没返回结果不计费；

1.5 空号检测 API 接口请求例子：

```
JAVA:  
public static main(String[] args){  
    Map<String,Object> params = new HashMap<>();
```

```
params.put("mobiles", "18610101065");
params.put("code", "YW28896484");
params.put("pwd", "JS78623352");
params.put("timestamp", "1557390636");
params.put("token", "13070ba6281d3c8006587276eec72f33");
System.out.println("api 空号检测 api 接口测试,请求参数:" + params.toString());
String responseStr =
HttpUtil.Post("https://kh.ywjisu.com/Api/ApiAuth/getCheckResult",params)
;
System.out.println("api 空号检测 api 接口测试,请求结果:" + responseStr1);
}
```

```
public static String post(String url, Map<String, Object> params) {
    try {
        CloseableHttpClient httpClient = HttpClients.createDefault();
        HttpPost httpPost = new HttpPost(url);
        List<NameValuePair> parameters = new ArrayList<NameValuePair>();
        for (Iterator<String> iterator = params.keySet().iterator(); iterator.hasNext();) {
            String key = iterator.next();
            if (params.get(key) != null) {
                parameters.add(new BasicNameValuePair(key,
params.get(key).toString()));
            }
        }
        UrlEncodedFormEntity uefEntity = new UrlEncodedFormEntity(parameters,
"UTF-8");
        httpPost.setEntity(uefEntity);
        CloseableHttpResponse response = httpClient.execute(httpPost);
        try {
            HttpEntity entity = response.getEntity();
            if (entity != null) {
                return EntityUtils.toString(entity, "UTF-8");
            }
        } finally {
            response.close();
        }
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
    return null;
}
```



PHP:

```
<?php
//post 方式
$curlPost=
"code=YW28896484&pwd=JS78623352&mobiles=15996789030&&timestamp=155739
2223&token=662ee2150b03e27ee566bb34d4877458";
$ch=curl_init();
curl_setopt($ch,CURLOPT_URL,'https://kh.ywjisu.com/Api/ApiAuth/getCheckRe
sult ');
curl_setopt($ch,CURLOPT_HEADER,0);
curl_setopt($ch,CURLOPT_RETURNTRANSFER,0);
//设置是通过 post
curl_setopt($ch,CURLOPT_POST,1);
//传递的变量
curl_setopt($ch,CURLOPT_POSTFIELDS,$curlPost);
$data = curl_exec($ch);
curl_close($ch);
?>
```