

# 接口管理平台

## 接口调用文档



## 目录

1. 文档介绍 .....	1
1.1. 文档目的 .....	1
1.2. 读者对象 .....	1
1.3. 术语与缩写解释 .....	1
2. 接口说明 .....	2
2.1. 接口调用入口接口 .....	2
3. 附录 .....	3
3.1. 附录 1 返回值状态码说明 .....	3

## 1. 文档介绍

### 1.1. 文档目的

接口管理平台接口调用文档。

### 1.2. 读者对象

在接口管理平台注册用户申请调用接口的相关单位开发人员。

### 1.3. 术语与缩写解释

无

## 2. 接口说明

### 2.1. 接口调用入口接口

接口	http://接口 IP:端口/interfaceExecute/openInterface/forward				
请求方式	post				
请求头参数	参数名	参数类型	默认值	是否必填	参数描述
	userKey	String		是	用户令牌，调用接口的唯一标识，可以从：《已申请接口-》申请详情-》详情；查看
	userId	String		否	用户 id，默认为接口调用申请人 id
请求参数	参数名	参数类型	默认值	是否必填	参数描述
	userGetSign	String		是/否	签名值，开启鉴权需要必填该项
请求参数说明	1. 请求参数除 userGetSign 外，其他参数根据各个注册接口的参数需求填写。				



返回值格式	JSON		
返回值参数	参数名	参数描述	
返回值说明	返回值，除报错信息外，将直接返回数据，返回数据格式详情见各注册接口的返回值格式		
返回值示例	<pre> {     "msg": "测试中文转发",     "test": true,     "data": {} } </pre>		

### 3. 附录

#### 3.1. 附录 1 返回值状态码说明

状态码	说明
-----	----

200	请求成功
401	请求异常
500	接口异常
501	请求异常

#### 1. 附录 4 HTTP 请求示例

```
/**
2. *
3. * @param url 接口地址（地址前需要加 http://）
4. * @param userKey 用户码，例：1729959225852411904
5. * @param secretKey 用户秘钥 例：1731278793592541184
6. * @param formData 请求参数
7. * 将请求参数和对应的值放到 map 中，各个参数对应 key 值请参考，表中请求参数对应值
8. * sign 对应的值需要根据接口方提供的秘钥和加密方法进行加密后,当做 value 放到 map 中
9. * @return
10. */
11. public static ResponseEntity
    sendPostQuery(String url,String userKey,String secretKey,Map<String, String> formData) {
12.     SimpleClientHttpRequestFactory requestFactory = new
        SimpleClientHttpRequestFactory();
13.     requestFactory.setConnectTimeout(5000);
14.     requestFactory.setReadTimeout(300000);
15.     RestTemplate restTemplate = new RestTemplate(requestFactory);
16.     try {
17.         /**
18.         * formData 中以键值对的形式存放请求参数和 sign
19.         */
20.         HttpHeaders headers = new HttpHeaders();
21.         headers.set("userKey",userKey);
22.         formData.put("type","JSON");
23.         formData.put("sign", getSign(formData,secretKey));
```

```
24. HttpEntity httpEntity = new HttpEntity(formData,headers);
25. ResponseEntity<String> response= restTemplate.postForEntity(url,
    httpEntity, String.class);
26. System.out.println(response.getBody());
27. return response;
28. } catch (Exception e) {
29. e.printStackTrace();
30. }
31. return null;
32.}
33./**
34. * 获取 sign 值
35. * @param map
36. * @param secretKry
37. * @return
38. */
39.private static String getSign(Map<String, String> map,String secretKry) {
40. Digester md5 = new Digester(DigestAlgorithm.SHA256);
41. String content = map.toString()+ "."+secretKry;
42. return md5.digestHex(content);
43.}
```