

04.元年税务管理平台开放接口规范-用户分册

修订记录

版本	修订人	修订日期	说明
v1.0	崔贵明	2018.09.05	起草

一、接口列表

序号	接口名称	接口描述
1.	用户授权	开发者对接税务管理平台，需要通过该接口获取平台授权，授权成功后获得Token（令牌）。消费其它平台服务时，需要提供Token。Token有效期有7200秒，不需要每次消费服务器获取Token，但需要在有效期内定时刷新。

二、接口规范

2.1 用户授权

开发者对接税务管理平台，需要通过该接口获取平台授权，授权成功后获得Token（令牌）。消费其它平台服务时，需要提供Token。Token有效期有7200秒，不需要每次消费服务器获取Token，但需要在有效期内定时刷新。

请求URL: https://host:port/openapi/user/token

请求类型: POST

2.1.1 公共参数:

名称	类型	是否必须	描述
appId	String	是	给开发者颁发的appId
appSecret	String	是	给开发者颁发的appSecret，appSecret采用加密传输，具体加密逻辑参看第四章
timestamp	String	是	获取授权时的时间戳，格式：yyyy-MM-ddTHH:mm:ss。时间戳与北京标准时间相差30分钟以上，禁止获取授权。标准北京时间参见： http://ntp.org/ ， NTP:cn.pool.ntp.org
format	String	是	响应报文格式，目前仅支持"JSON"
version	String	是	API协议版本，当前版本号"v1.0"

2.1.2 业务参数：

无

2.1.3 响应参数：

名称	类型	描述
accessToken	String	令牌
expiresIn	String	有效期（单位：秒）

2.1.4 SDK示例：

2.1.5 报文示例：

请求报文：

```
1 | {
2 |   "header":{
3 |     "appId":"100002",
4 |     "appSecret":"ecfadcde9305f8891bcfe5a1e28c1",
5 |     "timestamp":"2018-08-30T09:00:00",
6 |     "format":"JSON",
7 |     "version":"v1.0"
8 |   }
9 | }
```

响应报文：

```
1 | {
2 |     "code": 1000,
3 |     "message": "success",
4 |     "data": {
5 |         "accessToken": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiIxMDAwMDIiLCJzYWx0Ijoizj
6 |         "expiresIn": "7200"
7 |     }
8 | }
```

三、数据签名

3.1 签名算法

为保证业务数据在通信过程中不被篡改，需要服务消费方对业务参数进行数据签名。数据签名采用MD5算法对请求体JSON字符串加密生成（JSON字符串不包含“\r,\n”），MD5加密盐值为：“开发者盐值”。

3.2 签名算法代码

```

1  /**
2   * MD5加密
3   *
4   * @param value 明文
5   * @param salt 盐值
6   * @return
7   */
8  public static String md5Encrypt(String value, String salt) {
9      byte[] bytes = null;
10     try {
11         MessageDigest messageDigest = MessageDigest.getInstance("MD5");
12         messageDigest.update(salt.getBytes());
13         bytes = messageDigest.digest(value.getBytes("UTF-8"));
14
15         StringBuilder sign = new StringBuilder();
16         for (byte b : bytes) {
17             String hex = Integer.toHexString(b & 0xFF);
18             if (hex.length() == 1) {
19                 sign.append("0");
20             }
21             sign.append(hex.toUpperCase());
22         }
23         return sign.toString();
24     } catch (Exception e) {
25         e.printStackTrace();
26     }
27     return null;
28 }
29
30 public static void main(String[] args) {
31     String requestBodyJson = "{\"taxNo\":\"9111010875872474XN\", \"invoiceCode\"";
32     String appSalt = "ecfadcdc9305f8891bc";
33     System.out.println(md5Encrypt(requestBodyJson, appSalt));
34 }

```

四、密码(appSecret)加密

4.1 密码加密算法

开发者密码采用MD5加密传输，MD5加密盐值为："开发者盐值-获取授权时的时间戳"。

4.1 密码加密代码

```
1  /**
2   * MD5加密
3   *
4   * @param value 明文
5   * @param salt 盐值
6   * @return
7   */
8  public static String md5Encrypt(String value, String salt) {
9      byte[] bytes = null;
10     try {
11         MessageDigest messageDigest = MessageDigest.getInstance("MD5");
12         messageDigest.update(salt.getBytes());
13         bytes = messageDigest.digest(value.getBytes("UTF-8"));
14
15         StringBuilder sign = new StringBuilder();
16         for (byte b : bytes) {
17             String hex = Integer.toHexString(b & 0xFF);
18             if (hex.length() == 1) {
19                 sign.append("0");
20             }
21             sign.append(hex.toUpperCase());
22         }
23         return sign.toString();
24     } catch (Exception e) {
25         e.printStackTrace();
26     }
27     return null;
28 }
29
30 public static void main(String[] args) {
31     String appSecret = "ecfadcde9305f8891bcfe5a1e28c1u";
32     String appSalt = "ecfadcde9305f8891";
33     String timestamp = "2018-09-07T15:06:00";
34     System.out.println(md5Encrypt(appSecret, appSalt + "-" + timestamp));
35 }
```