|  |
| --- |
|  |

|  |
| --- |
|  |
| 接口说明-集成认证 |
| 版本 V1.0 |

**会签**

陕西 西安

2021．02

|  |  |  |  |
| --- | --- | --- | --- |
| 作者 | ZhengWei(HY) | 日期 | 2021-02-02 |
| 审核 |  | 日期 |  |
| 批准 |  | 日期 |  |

**版本**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 更新 | 签字 |
| 1.0 | 2021-02-02 | 创建 | ZhengWei(HY) |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目 录**

[1. 接口格式 5](#_Toc63281209)

[1.1 生成应用密钥对 5](#_Toc63281210)

[1.2 获取访问Token 6](#_Toc63281211)

[1.3 设置票据与登录用户的关系 8](#_Toc63281212)

# 接口格式

所有的 API 都是基于 HTTP 1.1 [RFC2616]， 采用 REST 风格定义。每一个请求都包含  
HTTP 的 POST、 GET 等请求方法，对应于请求的每一个响应都含有标准的 HTTP 状态  
码(status code)。大部分请求响应还包含了 JSON 格式的消息正文。

### 1.1 生成应用密钥对

此接口对于一个应用只调用一次就好，须将返回信息汇总管理员，并添加到ms.xsso.appKey.Config.xml配置文件中才能生效（请重启服务）。

#### 1.1.1 接口请求参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | 生成应用密钥对 | | | | |
| **说明** | 生成应用密钥对 | | | | |
| **地址** | <http://127.0.0.1:80/>服务名称/xsso/createApp | | | | |
| **方式** | Get请求 | | | | |
| **接口请求参数** | | | | | |
| **序号** | **参数名称** | **参数说明** | **参数类型** | **是否必须** | **备注** |
| 01 | 无 | 无 | 无 | 无 |  |

#### 1.1.2 接口响应参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **接口返回参数** | | | | | |
| **序号** | **参数名称** | **参数说明** | **参数类型** | **是否必须** | **备注** |
| 01 | code | 是否成功标识 | String | 是 | 200为成功 |
| 02 | message | 消息说明 | String | 是 |  |
| 03 | appKey | 应用编码 | String | 是 | 注册应用方妥善保存 |
| 04 | privateKey | 私钥加密验证 | String | 是 | 注册应用方妥善保存 |
| 05 | publicKey | 公钥加密验证 | String | 否 | 本服务妥善保存在：ms.xsso.appKey.Config.xml配置文件中。 |

#### 1.1.3 接口举例-请求报文

无

#### 1.1.4 接口举例-响应报文

{

    "code": "200",

    "message": "成功",

    "data": {

        "data": {

            "appKey": "70A78B5680FD4C7C9A9D9522F41F5E5E",

            "privateKey": "",

            "publicKey": "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQD0ry8BiVBTOg7APlYAwlmYpiqQAOYkje2BpDbtebfUow2LWxX1Gv8kSKO/u0A7CgD2NnI3JyKITUy2m+o/vXJ6qQ7aP1geOPz2FQV9uNvIFUcf1dCO1axASEMeJgWLhOhDikeymCevf7sv/2PMzd80xvFFnpQdv9xTmX4wdSu1sQIDAQAB"

        }

    }

}

### 1.2 获取访问Token

应用创建成功后，会生成应用的 appKey 和 appSecret 的值，租户管理员可将这个两个值交给开发者，开发者可用 appKey 和 appSecret 获取 token 进行接口的调用。

#### 1.2.1 接口请求参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | 获取访问Token | | | | |
| **说明** | 调用接口令牌 access\_token 是应用调用开放平台业务接口的凭证，有效期为2小时,过期后需要重新获取。 | | | | |
| **地址** | <http://127.0.0.1:80/>服务名称/xsso/createApp | | | | |
| **方式** | Get请求 | | | | |
| **接口请求参数** | | | | | |
| **序号** | **参数名称** | **参数说明** | **参数类型** | **是否必须** | **备注** |
| 01 | appKey | 应用 appKey | String | 是 |  |
| 02 | timestamp | 毫秒时间戳 | Long | 是 |  |
| 03 | signature | 校验签名 | String | 是 |  |

**signature校验签名的生成方式**

其中，签名字段signature计算使用RAS算法，具体计算方式如下：

URLEncode( Base64EnCode( RAS( parameters) ) )

其中，parameters 按照参数名称排序，参数名称与参数值依次拼接(signature字段除外)，形成待计算签名的字符串。

示例 ：若发送请求参数appKey为0904202FB2EA4AC3A899B423928A0F8E,时间戳timestamp为1609221942639，则待加密字符串的值为 appKey0904202FB2EA4AC3A899B423928A0F8E timestamp1609221942639

之后对 parameters使用 RAS 计算签名，RAS 的 key 为自建应用的 appSecret 。计算出的二进制签名先进行 Base64，之后进行 URLEncode，即得到 signatrue 字段的值。

#### 1.2.2 接口响应参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **接口返回参数** | | | | | |
| **序号** | **参数名称** | **参数说明** | **参数类型** | **是否必须** | **备注** |
| 01 | code | 是否成功标识 | String | 是 | 200为成功 |
| 02 | message | 消息说明 | String | 是 |  |
| 03 | accessToken | 访问令牌 | String | 否 | 成功时，返回 |
| 04 | expire | 过期时长 | String | 否 | 成功时，返回。  单位：秒  默认为：7200 |
| 05 | code | 临时登录Code | String | 否 | 成功时，返回。  在其后的使用中，仅且只能使用一次，做为其它接口的入参 |

#### 1.2.3 接口举例-请求报文

<http://127.0.0.1/>msXSSO/**getAccessToken**?**appKey**=0904202FB2EA4AC3A899B423928A0F8E&**timestamp**=1609221942639&**signature**=fPAHpZghqNLpssMr%2FeMlbTizQSpCX91hRMRKdrKs0djDCyH4Hb0ozv7yfFSgjGBTGLhd6K%2FAqxE3UQQMmk33OCiuHztNbyRaxSGgrQ75HIVmvriQFYtPTUGOtYR4vmKzGOcxghv4s%2FTdWsGGRWwcK8y%2Bg9ZULRHVUtu8EMErud4%3D

#### 1.2.4 接口举例-响应报文

{

    "code": "200",

    "message": "成功",

    "data": {

        "data": {

            "accessToken": "B8EE43B314E94A5CBC46F7C266043FFA",

            "expire": 7200,

            "code": "51FDA1E47E794AE9A2ED2A51E9B001C7"

        }

    }

}

### 1.3 设置票据与登录用户的关系

#### 1.3.1 接口请求参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | 设置票据与登录用户的关系 | | | | |
| **说明** | 用于用户成功登录后，将票据与用户信息绑定。 | | | | |
| **地址** | <http://127.0.0.1:80/>服务名称/xsso/setLoginUser | | | | |
| **方式** | Get请求 | | | | |
| **接口请求参数** | | | | | |
| **序号** | **参数名称** | **参数说明** | **参数类型** | **是否必须** | **备注** |
| 01 | code | 临时登录Code | String | 是 | Code使用后失效，仅能使用一次。 |
| **接口请求体** | | | | | |
| 01 | appKey | 应用编码 | String | 是 |  |
| 02 | userId | 用户在第三方用户中心的标识 | String | 是 |  |
| 03 | userNo | 用户在第三方用户中心的工号 | String | 否 |  |
| 04 | userCode | 用户在第三方系统的编码 | String | 否 |  |
| 05 | openID | 微信用户应用ID | String | 否 |  |
| 06 | unionid | 微信用户统一ID | String | 否 |  |
| 07 | mobile | 用户在第三方系统的手机号 | String | 否 |  |
| 08 | phone | 用户在第三方系统的手机号 | String | 否 |  |
| 09 | email | 用户在第三方系统的邮箱 | String | 否 |  |
| 10 | userName | 用户在第三方系统的用户名 | String | 否 |  |
| 11 | nickname | 用户在第三方系统的昵称 | String | 否 |  |
| 12 | userType | 用户在第三方系统的用户类型 | String | 否 |  |
| 13 | userLevel | 用户在第三方系统的用户级别 | String | 否 |  |
| 14 | userSource | 户在第三方系统的用户来源 | String | 否 |  |
| 15 | loginAccount | 用户在第三方系统的登陆账号 | String | 否 |  |
| 16 | orgCode | 用户在第三方系统的组织编码 | String | 否 |  |
| 17 | orgName | 用户在第三方系统的组织名称 | String | 否 |  |
| 18 | sex | 用户性别 | String | 否 |  |
| 19 | sessionID | 会话ID | String | 否 |  |
| 20 | datas | 附加用户数据 | Map  <String ,String> | 否 |  |

#### 1.3.2 接口响应参数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **接口返回参数** | | | | | |
| **序号** | **参数名称** | **参数说明** | **参数类型** | **是否必须** | **备注** |
| 01 | code | 是否成功标识 | String | 是 | 200为成功 |
| 02 | message | 消息说明 | String | 是 |  |
| 03 | expire | 过期时长 | Integer | 否 | 成功时返回 |

#### 1.3.3 接口举例-请求报文

http://127.0.0.1/msXSSO/xsso/setLoginUser?code=3E8DE247F5E84EC592ECF633807BF3A2

{

    "appKey": "D22DCADC06BB4EAE868D5728AACAD370",

    "userId": "9527",

    "userNo": "工号",

    "userCode": "编码",

    "mobile": "手机号",

    "phone": "手机号",

    "email": "邮箱",

    "userName": "用户名",

    "nickname": "昵称",

    "userType": "用户类型",

    "userLevel": "用户级别",

    "userSource": "用户来源",

    "loginAccount": "登陆账号",

    "orgCode": "组织编码",

    "orgName": "组织名称",

    "sex": "用户性别",

    "sessionID": "会话ID",

    "datas": {

        "附加用户数据1": "附加用户数据1",

        "附加用户数据2": "附加用户数据2"

    }

}

#### 1.3.4 接口举例-响应报文

{

    "code": "200",

    "message": "成功",

    "data": {

        "data": {

            "expire": 7177

        }

    }

}