|  |
| --- |
| 上海宁盾信息科技有限公司 |
| DKEY AM API |
| 二次开发文档 |

|  |
| --- |
| 宁盾科技  05/04/2016 |

目录

[目录 2](#_Toc516740786)

[1. 认证 3](#_Toc516740787)

[1.1. Authenticate 3](#_Toc516740788)

[1.2. 获取密码使用（令牌绑定）情况 4](#_Toc516740789)

[1.3. 发送动态密码到手机 5](#_Toc516740790)

[2. 站点 5](#_Toc516740791)

[2.1. 获取站点 5](#_Toc516740792)

[3. 用户 6](#_Toc516740793)

[3.1. 外部用户认证 6](#_Toc516740794)

[3.2. 获取单个用户信息 7](#_Toc516740795)

[3.3. 同步所有用户 9](#_Toc516740796)

[3.4. 增量同步用户（创建或更新） 12](#_Toc516740797)

[3.5. 设定用户手机号 13](#_Toc516740798)

[3.6. 删除用户 14](#_Toc516740799)

[3.7. 解绑定用户终端 14](#_Toc516740800)

[3.8. 申请临时用户（通过手机或邮箱告知密码） 15](#_Toc516740801)

[3.9. 查询双因素用户的登录情况 16](#_Toc516740802)

[4. 令牌 16](#_Toc516740803)

[4.1. 派发手机令牌（单发） 16](#_Toc516740804)

[4.2. 获取手机令牌激活二维码 17](#_Toc516740805)

[4.3. 验证时间型令牌密码 18](#_Toc516740806)

[4.4. 绑定时间型令牌 18](#_Toc516740807)

[4.5. 解绑时间型令牌 19](#_Toc516740808)

[4.6. 解绑所有时间型令牌 20](#_Toc516740809)

[4.7. 令牌和用户的绑定关系 20](#_Toc516740810)

[4.8. 令牌使用概况 21](#_Toc516740811)

[4.9. 认证系统令牌列表 21](#_Toc516740812)

[5. Portal 22](#_Toc516740813)

[5.1. 增量更新用户信息（微信） 22](#_Toc516740814)

[5.2. 第三方验证码登录验证（微信） 22](#_Toc516740815)

[5.3. 用户登录前获取外部角色（短信，微信） 23](#_Toc516740816)

[5.4. 断开在线用户 24](#_Toc516740817)

[5.5. APP扫码授权（协助扫码） 26](#_Toc516740818)

[6. 附录 27](#_Toc516740819)

[6.1. 请求响应码 27](#_Toc516740820)

[6.2. 数据结构 27](#_Toc516740821)

[6.2.1. TenantInfo 27](#_Toc516740822)

[6.2.2. StrongAuthSession 27](#_Toc516740823)

[6.2.3. TokenInfo 28](#_Toc516740824)

[6.2.4. UserInfo 28](#_Toc516740825)

[6.2.5. TokenUsage 28](#_Toc516740826)

[6.2.6. TokenBinding 28](#_Toc516740827)

# 认证

## Authenticate

**说明：**根据登录名和动态码校验身份合法性

**URL:** http(s)://ip:port/am/webauth/1/strong/authenticate

**请求方式：**GET 或 POST

**编码：**所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**参数：**

* **tenantName（必填）站点**名称
* **accessServerName（必填）**认证设备名称
* **sharedSecret（必填）**认证设备共享密钥“
* **loginName（必填）**登录名
* **password（必填）**密码

**JSON返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 是否成功 | true: 成功  false: 失败 |
| errorCode | int | 错误类型 | -16：用户名或密码错误  -17：用户被锁定  -3：参数不正确  -10：认证服务器配置错误  -1001：用户被禁用或过期 |
| roles | JSON Array | 用户角色列表 | 如["group1","group2"] |
| message | string | 错误消息 |  |

**示例：**

[http(s)://192.168.1.30:8080/am/webauth/1/strong/authenticate?tenantName=mytenant&accessServerName=myoa&sharedSecret=123456&loginName=testUser&password=testPassword](http://192.168.1.30:8080/am/webauth/1/strong/authenticate?tenantName=mytenant&accessServerName=myoa&sharedSecret=123456&loginName=testUser&password=testPassword)

示例返回值：

成功：

{"success":true,"errorCode":0,"dynamicPasswordLength":0,"message":null,"challenging":false,"sessionId":null,"attributes":null,"roles":["111","Guest"]}

失败：

{"success":false,"errorCode":-16,"dynamicPasswordLength":0,"message":"用户名或密码错误.","challenging":false,"sessionId":null,"attributes":null,"roles":["111","Guest"]}

## 获取密码使用（令牌绑定）情况

**说明：**查看用户是否需要动态密码（是否需要动态密码框），是否需要通过短信发送动态码

**URL:** http(s)://ip:port/am/webauth/1/strong/passwordRequirement

**请求方式：**GET 或 POST

**编码：**所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**参数：**

* **tenantName（必填）站点**名称
* **accessServerName（必填）**认证设备名称
* **sharedSecret（必填）**认证设备共享密钥
* **loginName（必填）**用户名

**JSON返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 是否成功 | true: 成功  false: 失败 |
| errorCode | int | 错误类型 | -3：参数不正确  -10：认证服务器配置错误 |
| message | string | 错误消息 |  |
| data | JSON | 密码使用情况 |  |
| **data中的字段** | | | |
| staticPasswordRequired | boolean | 是否需要静态密码 |  |
| dynamicPasswordRequired | boolean | 是否需要动态密码 |  |
| needsGetDynamicPassword | boolean | 是否需要调用getDynamicPassword（如果用户绑定了短信令牌，则返回true） |  |

**示例：**

http(s)://192.168.1.30:8080/am/webauth/1/strong/passwordRequirement?tenantName=mytenant&accessServerName=myoa&sharedSecret=123456&loginName=testUser

示例返回值：

{

"success" : true,

"data" :

{

"staticPasswordRequired" : false,

"dynamicPasswordRequired" : true,

"needsGetDynamicPassword" : false

}

}

## 发送动态密码到手机

**说明：**如果1.2接口中dynamicPasswordRequired=true并且needsGetDynamicPassword=true，说明用户有短信开启了动态密码，并且有短信令牌，调用此接口，将动态码发送到手机

**URL:** http(s)://ip:port/am/webauth/1/strong/getDynamicPassword

**请求方式：**GET 或 POST

**编码：**所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**参数：**

* **tenantName（必填）**站点名称
* **accessServerName（必填）**认证设备名称
* **sharedSecret（必填）**认证设备共享密钥
* **loginName（必填）**登录名

**JSON返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 是否成功 | true: 成功  false: 失败 |
| errorCode | int | 错误类型 | -3：参数不正确  -10：认证服务器配置错误 |
| message | string | 错误消息 |  |

**示例：**

http(s)://192.168.1.30:8080/am/webauth/1/strong/getDynamicPassword?tenantName=mytenant&accessServerName=myoa&sharedSecret=123456&loginName=testUser

示例返回值：

{"success":false,"errorCode":0,"message":"短信发送失败."}

# 站点

## 获取站点

**URL：**http(s)://ip:port/am/rest/p/1/tenant/list

**编码：**所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**参数：**

* **adminLoginName（必填）**管理员登录名
* **adminPassword（必填）**管理员密码

返回值：

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 是否成功 | true: 成功  false: 失败 |
| message | String | 错误消息 | 可以为空 |
| errorCode | int | 结果的代码 | 如果成功，errorCode为0 |
| data | List<TenantInfo> | 站点的集合 | 引用6.2.1:TenantInfo |

# 用户

## 外部用户认证

**说明：**去第三方系统校验静态密码

**配置：**



**URL：**第三方系统提供

**协议：**HTTP(S)

**请求方式：**GET 或 POST

**参数：**

* **loginName（必填，String）**登录名
* **password（必填，String）**密码
* **secretKey（可选，String）**接口调用的密钥

**JSON返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 是否成功 | true: 成功  false: 失败 |
| message | String | 错误消息 | 可以为空 |
| errorCode | int | 结果代码 | 如果成功，errorCode为0 |
| data | json | 成功后，用户信息放在data里面，可以只填写其中的一部分 |  |
| data: personalName | String | 用户信息的个人姓名 | 可以不含这个字段 |
| data: loginName | String | 用户信息的登录名字 | 可以不含这个字段 |
| data: role | String | 用户的角色 | 可以不含这个字段 |
| data: email | String | 用户的邮箱 | 可以不含这个字段 |
| data: mobile | String | 用户的手机 | 可以不含这个字段 |

**示例：**

<http://192.168.1.10:8080/crm/user/auth?loginName=testUser&password=testPassword>

示例返回值：

示例一：

{

"success": false,

"errorCode": -1,

"data": null,

"message": "用户名或密码错误."

}

示例二：

{

"success": true,

"errorCode": 0,

"data": {

"loginName": "testUser",

"role": "admin",

"email": "testUser@test.com",

"mobile": "13812345678"

},

"message":null

}

## 获取单个用户信息

**说明：**从第三方系统获取用户信息

**配置：**



**URL：**第三方系统提供

**协议：**HTTP(S)

**调用方式：**GET 或POST

**参数：**

* **loginName（必填，String）**登录名
* **secretKey（可选，String）**接口调用的密钥

**JSON返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 是否成功 | true: 成功  false: 失败 |
| message | String | 错误消息 | 可以为空 |
| errorCode | int | 结果代码 | 如果成功，errorCode为0 |
| data | json | 成功后，用户信息放在data里面，可以只填写其中的一部分 |  |
| data: personalName | String | 用户信息的个人姓名 | 可以不含这个字段 |
| data: loginName | String | 用户信息的登录名字 | 可以不含这个字段 |
| data: role | String | 用户的角色 | 可以不含这个字段 |
| data: email | String | 用户的邮箱 | 可以不含这个字段 |
| data: mobile | String | 用户的手机 | 可以不含这个字段 |

**示例：**

<http://192.168.1.10:8080/crm/user/userInfo?loginName=xiaoming&secretKey=sowososoed>

示例返回值：

示例一：

{

"success": false,

"errorCode": -1,

"data": null,

"message": "不存在该用户."

}

示例二：

{

"success": true,

"errorCode": 0,

"data": {

"loginName": "xiaoming",

"role": "admin",

"email": " xiaoming@test.com",

"mobile": "13812345678"

},

"message":null

}

## 同步所有用户

**说明：**宁盾认证系统调用第三方系统的接口，把第三方系统的用户信息全部同步到认证系统

**配置：**



**URL:** 第三方系统提供

**协议：**HTTP(S)

**请求方式：**GET 或 POST

**参数：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数** | **是否必须** | **类型** | **定义** | **值** |
| startRow | 否 | int | 如果通过分页的方式实现同步，startRow表示第几页的数据 | 从0开始， startRow = 0表示第一页 |
| limit | 否 | int | 如果通过分页的方式实现同步，limit表示每页的数据量。例如，startRow = 0，limit = 200，表示请求第1页的200条数据 | 必须大于0 |

请求参数说明：

第三方系统可以使用分页或者不分页的方式实现该接口。如果使用分页的方式（即接口每次调用只是获取部分用户的信息），那么需要处理startRow和limit两个参数, startRow从0开始计数，limit必须大于0，例如， startRow = 0 & limit = 10， 取第1条到第10条客户数据， startRow = 1 & limit = 10, 取第11条到第20条客户数据，以此类推；如果不使用分页的方式（即接口每次调用获取的是全部用户的信息）,请忽略startRow和limit参数，返回所用用户的信息。

**JSON返回值（返回用户的信息）：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **变量** | **是否必须返回** | **类型** | **定义** | **值** |
| success | 是 | boolean | 调用同步数据接口是否成功 | true: 成功  false: 失败 |
| errorCode | 是 | int | 调用同步数据接口返回的errorCode | 由第三方接口定义，默认errorCode=0表示调用成功 |
| message | 是 | String | 调用同步数据接口返回的信息 | 可以返回null |
| data: totalCount | 是 | int | 需要同步的用户的人数 | 不管是否实现分页，totalCount都是表示所有需要同步的用户的人数 |
| data:data:loginName | 是 | String | 用户通过认证系统所使用的登陆名称 |  |
| data:data:personalName | 否 | String | 用户姓名 |  |
| data:data:role | 否 | String | 用户角色 |  |
| data:data:email | 否 | String | 用户邮箱 |  |
| data:data:mobile | 否 | String | 用户手机号码 |  |
| data:data:enabled | 否 | boolean | 用户是否启用 | 说明：如果该值不填或者为null，默认是启用；  true: 启用该用户  false：禁用该用户 |
| data:data:startTime | 否 | long | 用户启用的时间戳，Unix timestamp, 单位：毫秒 | 如果data:data:enabled的值已经填写，请务必填写该项返回值 |
| data:data:endTime | 否 | long | 用户启用的时间戳，Unix timestamp, 单位：毫秒 | 如果data:data:enabled的值已经填写，请务必填写该项返回值 |

JSON返回值示例：

情景一：接口实现了分页

调用url：http://test.com:8080/syncAllUsers

请求参数： startRow = 0, limit = 2

返回json的信息：

{

"success": true,

"errorCode": 0,

"message": null,

"data": {

"totalCount": 4,

"data": [

{

"loginName": "xiaoming",

"personalName": "小明",

"role": "admin",

"email": " xiaoming@test.com",

"mobile": "13812345678"

},

{

"loginName": " zhangsan ",

"personalName": "张三",

"role": "employee",

"email": " zhangsan@test.com",

"mobile": "13687654321",

"enabled": true,

"startTime": 1479484801000,

"endTime": 1511020801000

}

]

}

}

情景二：接口没有实现分页，忽略请求参数

调用url：http://test.com:8080/syncAllUsers

请求参数： startRow = 0, limit = 2

返回json的信息：

{

"success": true,

"errorCode": 0,

"message": null,

"data": {

"totalCount": 4,

"data": [

{

"loginName": "xiaoming",

"personalName": "小明",

"role": "admin",

"email": " xiaoming@test.com",

"mobile": "13812345678"

},

{

"loginName": " zhangsan ",

"personalName": "张三",

"role": "employee",

"email": " zhangsan@test.com",

"mobile": "13687654321"

},

{

"loginName": " lisi",

"personalName": "李四",

"role": "employee",

"email": " lisi@test.com",

"mobile": "13398765432"

},

{

"loginName": " wangwu ",

"personalName": "王五",

"role": "employee",

"email": " wangwu@test.com",

"mobile": "13611111111",

"enabled": true,

"startTime": 1479484801000,

"endTime": 1511020801000

}

]

}

}

## 增量同步用户（创建或更新）

**说明：**在认证系统上创建用户，或更新用户

**URL:** http(s)://ip:port/am/rest/p/1/identity/user/incSync

**请求方式： GET 或 POST**

**编码**：所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**参数：**

* **adminLoginName（必填）**管理员登录名
* **adminPassword（必填）**管理员密码
* **tenantName（必填）**站点名称
* **identityStoreName（必填）**用户库名称。本地用户库名称为06bc758e-04f2-40b3-9d6c-562b115aeb3c，外部用户库名称和创建的外部用户库一致
* **loginName（必填）**用户登录名
* **groupPath（可选）**组路径。组与组之间用 / 分开，组名支持 中文！英文 . 数字-\_
* **personalName （可选）**用户姓名
* **password（可选）**用户密码
* **needPasswordPolicy（可选，Boolean，默认true）**密码策略，密码复杂度
* **mobile（可选）**用户手机号
* **roleName（可选，注意roleName和roleNames不可同时使用）**用户角色
* **roleNames （可选）**多个角色
* **startTime（可选，Long，毫秒，注意startTime和endTime必须同时有，或同时为空）**生效时间，Unix Timestamp
* **endTime（可选，Long，毫秒）**失效时间，Unix Timestamp

**特殊说明：**

**roleName** 每次更新一个角色，不会解绑以前的角色

**roleNames** 每次可以更新多个角色，会将以前的角色解绑；示例&roleNameList=["aa","bbc"]

如果url中没有这个参数名，则不修改

**groupPath** 更新用户，如果在url中没有这个参数名，则不修改

**JSON返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 操作是否成功 | true: 成功  false: 失败 |
| message | string | 错误消息 |  |

**示例（在本地用户创建或更新testUser，密码123456，手机号**13800000000**）：**

http://192.168.1.30:8080/am/rest/p/1/identity/user/incSync?tenantName=TestTenant&identityStoreName=06bc758e-04f2-40b3-9d6c-562b115aeb3c&groupPath=1/2/3&loginName=testUser&password=123456&mobile=13800000000&roleName=&startTime=1452999990000&endTime=1453086390000&adminLoginName=admin&adminPassword=admin



示例返回值：

{"success":true,"errorCode":0,"data":null,"message":null}

## 设定用户手机号

**说明：**修改用户手机号

**URL:** http(s)://ip:port/am/rest/p/1/identity/user/mobile/set

**请求方式： GET 或 POST**

**编码**：所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**参数：**

* **adminLoginName（必填）**管理员登录名
* **adminPassword（必填）**管理员密码
* **tenantName（必填）**站点名称
* **identityStoreName（必填）**用户库名称。本地用户库名称为06bc758e-04f2-40b3-9d6c-562b115aeb3c，外部用户库名称和创建的外部用户库一致
* **loginName（必填）**用户登录名
* **mobile（必填）**用户手机号

**JSON返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 操作是否成功 | true: 成功  false: 失败 |
| message | string | 错误消息 |  |

**示例：**

<http://192.168.1.30:8080/am/rest/p/1/identity/user/mobile/set?tenantName=TestTenant&identityStoreName=TestAD&loginName=testUser&mobile=13800000000&adminLoginName=admin&adminPassword=admin>

示例返回值：

{"success":true,"errorCode":0,"data":null,"message":null}

## 删除用户

**URL:** <http://ip:port/am/rest/p/1/identity/user/delete>

**编码：**所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**参数：**

* **tenantName（必填）**: 商户名称；
* **identityStoreName（必填）**: 用户库名称；
* **loginName（必填）**: 用户登录名；
* **adminLoginName（必填）**: 管理员用户名；
* **adminPassword（必填）**: 管理员密码。

**JSON返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 操作是否成功 | true: 成功  false: 失败 |
| message | string | 错误消息 |  |

**示例：**

http://192.168.1.30:8080/am/rest/p/1/identity/user/delete?tenantName=TestTenant&identityStoreName=TestAD&loginName=testUser&adminLoginName=admin&adminPassword=admin

示例返回值：

{"success":true,"errorCode":0,"data":null,"message":null}

## 解绑定用户终端

**URL:** http://ip:port/am/rest/p/1/identity/user /unbindUserTerminals

**编码：**所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**参数：**

* **tenantName（必填）**: 商户名称；格式：String；
* **identityStoreName（必填）**: 用户库名称；格式：String；
* **loginName（必填）**: 用户登录名；格式：String；
* **adminLoginName（必填）**: 管理员用户名；格式：String；
* **adminPassword（必填）**: 管理员密码；格式：String；
* **deleteFingerprint（选填）:** 是否删除指纹库的MAC地址； 格式boolean

**JSON返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 操作是否成功 | true: 成功  false: 失败 |
| message | string | 错误消息 |  |

**示例1：**

http://192.168.1.30:8080/am/rest/p/1/identity/user/unbindUserTerminals?tenantName=TestTenant&identityStoreName=TestAD&loginName=testUser&adminLoginName=admin&adminPassword=admin

**示例2：**

http://192.168.1.30:8080/am/rest/p/1/identity/user/unbindUserTerminals?tenantName=TestTenant&identityStoreName=TestAD&loginName=testUser&adminLoginName=admin&adminPassword=admin&deleteFingerprint=true

示例返回值1：

{"success":true,"errorCode":0,"data":null,"message":null}

示例返回值2：

{"success":false, "errorCode":null, "data":null, "message": "User does not exist"}

## 申请临时用户（通过手机或邮箱告知密码）

**说明：**申请临时用户，并把登录名和密码发送至手机号或邮箱

**URL：**http(s)://ip:port/am/rest/p/1/identity/user/applyTemporaryUser

**编码：**所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**参数：**

* **adminLoginName（必填）**管理员登录名
* **adminPassword（必填）**管理员密码
* **tenantName（必填）**站点名称
* **identityStoreName（必填）**用户库名称（内部申请人所在用户库）。本地用户库名称为06bc758e-04f2-40b3-9d6c-562b115aeb3c，外部用户库名称和创建的外部用户库一致
* **applierLoginName（必填）内部申请人登录名，需要提前配置**
* **sendToApplier（boolean可选）发给申请人邮箱（默认false）**
* **sendToTemporaryUser（boolean可选）发给临时用户邮箱（默认true）**
* **sendAccountTo（必填，int类型）1：发送到手机号，2：发送到邮箱（默认发给临时用户）**
* **mobile（发送到手机时必填，并且作为登录账号）手机号**
* **email（发送到邮箱时必填，并且作为登录账号）邮箱**
* **startTime（必填，Long毫秒）开始时间**
* **endTime（必填，Long毫秒）结束时间**
* **roleName（可选）角色名,必须是在认证系统中配置的**
* **personalName（可选）姓名**
* **number（可选）证件号码（工号，护照等）**
* **idNumber（可选）身份证号码**
* **company（可选）公司名称**
* **description（可选）描述**
* **JSON返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 操作是否成功 | true: 成功  false: 失败 |
| message | string | 错误消息 |  |

**示例：**

**http://127.0.0.1:8080/am/rest/p/1/identity/user/applyTemporaryUser?adminLoginName=admin&adminPassword=admin&tenantName=test&identityStoreName=test&applierLoginName=0&sendAccountTo=1&mobile=13011110000&email=3135880854@qq.com&startTime=1524326400000&endTime=1524585600000**

**成功的返回：**

**{"success": true, "errorCode": 0, "data": null, "message": null}**

## 查询双因素用户的登录情况

**说明：**派发新令牌，默认解绑以前的令牌

**URL：**http(s)://ip:port/am/rest/p/1/accessSession/strongAuth/list

**编码：**所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**参数：**

* **adminLoginName（必填）**管理员登录名
* **adminPassword（必填）**管理员密码
* **tenantName（必填）**站点名称
* **startTime：（long毫秒，必填）**开始时间
* **endTime：（long毫秒，必填）**结束时间

**返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 是否成功 | true: 成功  false: 失败 |
| message | String | 错误消息 | 可以为空 |
| errorCode | int | 结果的代码 | 如果成功，errorCode为0 |
| data | List<StrongAuthSession> | 登录日志的集合 | 引用6.2.2:StrongAuthSession |

# 令牌

## 派发手机令牌（单发）

**说明：**派发新令牌，默认解绑以前的令牌

**URL:** http(s)://ip:port/am/rest/p/1/token/mobileToken/deliver

**编码：**所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**参数：**

* **adminLoginName（必填）**管理员登录名
* **adminPassword（必填）**管理员密码
* **tenantName（必填）**站点名称
* **identityStoreName（必填）**用户库名称
* **loginName（必填）**用户登录名
* **mode（必填）**派发模式。0表示派发到邮箱，1表示派发到手机短信
* **toEmail（可选）**激活码接收邮箱。如果不填写，激活码发送到用户默认邮箱中；如果用户没有默认邮箱，则必须填写
* **toMobile（可选）**激活码接收手机号。如果不填写，激活码发送到用户默认手机号中如果用户没有默认手机号，则必须填写
* **unbindPrevious（可选,Boolean,默认true）**是否解绑以前的令牌

**JSON返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 操作是否成功 | true: 成功  false: 失败 |
| message | string | 错误消息 |  |

**示例：**

<http://192.168.1.30:8080/am/rest/p/1/token/mobileToken/deliver?tenantName=TestTenant&identityStoreName=TestAD&loginName=testUser&mode=1&toMobile=13800000000&adminLoginName=admin&adminPassword=admin>

示例返回值：

{"success":true,"errorCode":0,"data":null,"message":null}

## 获取手机令牌激活二维码

**URL:** http(s)://ip:port/am/rest/p/1/token/mobileToken/userQrCode

**编码：**所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**参数：**

* **adminLoginName（必填）**管理员登录名
* **adminPassword（必填）**管理员密码
* **tenantName（必填）站点**名称
* **identityStoreName（必填）**用户库名称
* **loginName（必填）**用户登录名

**返回值：**二维码图片

**示例：**

<http://192.168.1.30:8080/am/rest/p/1/token/mobileToken/userQrCode?tenantName=TestTenant&identityStoreName=TestAD&loginName=testUser&adminLoginName=admin&adminPassword=admin>

示例返回值：



## 验证时间型令牌密码

**URL:** http(s)://ip:port/am/rest/p/1/token/verifyPassword

**编码：**所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**请求方式：**GET 或 POST

**参数：**

* **adminLoginName（必填）**管理员登录名
* **adminPassword（必填）**管理员密码
* **tenantName（必填）**站点名称
* **serial（必填）**令牌序号
* **password（必填）**令牌动态码

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 是否成功 | true: 成功  false: 失败 |
| message | String | 错误消息 | 可以为空 |
| errorCode | int | 结果的代码 | 如果成功，errorCode为0 |
| data | json | 成功后，用户信息放在data里面，可以只填写其中的一部分 | 0 成功  1站点不存在  2令牌不存在  3令牌动态码错误 |

## 绑定时间型令牌

**URL:** http://ip:port/am/rest/p/1/token/bind

**编码：**所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**请求方式：**GET 或POST

**参数：**

* **adminLoginName（必填）**管理员登录名
* **adminPassword（必填）**管理员密码
* **tenantName（必填）**站点名称
* **identityStoreName（必填）**用户库名称。本地用户库名称为06bc758e-04f2-40b3-9d6c-562b115aeb3c，外部用户库名称和创建的外部用户库一致
* **loginName（必填）**用户登录名
* **serial（必填）**令牌序号

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 是否成功 | true: 成功  false: 失败 |
| message | String | 错误消息 | 可以为空 |
| errorCode | int | 结果的代码 | 如果成功，errorCode为0 |
| data | json | 成功后，用户信息放在data里面，可以只填写其中的一部分 | 0 成功  1站点不存在  2用户库不存在  3用户不存在  4令牌不存在  6系统授权不合法  7此令牌已绑定其他用户 |

## 解绑时间型令牌

**URL:** http://ip:port/am/rest/p/1/token/unbind

**编码：**所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**请求方式：**GET 或 POST

**参数：**

* **adminLoginName（必填）**管理员登录名
* **adminPassword（必填）**管理员密码
* **tenantName（必填）站点名称**
* **identityStoreName（必填）**用户库名称。本地用户库名称为06bc758e-04f2-40b3-9d6c-562b115aeb3c，外部用户库名称和创建的外部用户库一致
* **loginName（必填）**用户登录名
* **serial（必填）**令牌序号

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 是否成功 | true: 成功  false: 失败 |
| message | String | 错误消息 | 可以为空 |
| errorCode | int | 结果的代码 | 如果成功，errorCode为0 |
| data | json | 成功后，用户信息放在data里面，可以只填写其中的一部分 | 0 成功  1站点不存在  2用户库不存在  3用户不存在  4令牌不存在 |

## 解绑所有时间型令牌

**URL:** http://ip:port/am/rest/p/1/token/unbindAll

**编码：**所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**请求方式：**GET 或 POST

**参数：**

* **adminLoginName（必填）**管理员登录名
* **adminPassword（必填）**管理员密码
* **tenantName（必填）**站点名称
* **identityStoreName（必填）**用户库名称。本地用户库名称为06bc758e-04f2-40b3-9d6c-562b115aeb3c，外部用户库名称和创建的外部用户库一致
* **loginName（必填）**用户登录名

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 是否成功 | true: 成功  false: 失败 |
| message | String | 错误消息 | 可以为空 |
| errorCode | int | 结果的代码 | 如果成功，errorCode为0 |
| data | json | 成功后，用户信息放在data里面，可以只填写其中的一部分 | 0 成功  1站点不存在  2用户库不存在  3用户不存在 |

## 令牌和用户的绑定关系

**说明：****从认证系统获取某个令牌与用户的绑定关系或者某个用户与令牌的绑定关系**

**URL:** http://ip:port/am/rest/p/1/token/binding/list

**请求方式：**GET 或 POST

**参数：**

* **adminLoginName（必填）**管理员登录名
* **adminPassword（必填）**管理员密码
* **tenantName(必填)** 站点名称

**查找方式二选一：**

1.根据用户查找：

* **identityStoreName（必填）**用户库名称。本地用户库名称为06bc758e-04f2-40b3-9d6c-562b115aeb3c，外部用户库名称和创建的外部用户库一致
* **loginName（必填）**用户登录名

2.根据令牌查找：

* **type(必填**)令牌类型**（1.时间令牌，2.短信令牌）**
* **serial(必填)**令牌序列号

**返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 是否成功 | true: 成功  false: 失败 |
| message | String | 错误消息 | 可以为空 |
| errorCode | int | 结果的代码 | 如果成功，errorCode为0 |
| data | List<TokenBinding> | 绑定关系集合 | 引用6.2.6:TokenBinding |

## 令牌使用概况

**说明：****获取令牌使用，令牌占用情况（目前系统授权按最近30天计算）**

**URL:** http://ip:port/am/rest/p/1/token/usage

**请求方式：**GET 或 POST

**参数：**

* **adminLoginName（必填）**管理员登录名
* **adminPassword（必填）**管理员密码

**返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 是否成功 | true: 成功  false: 失败 |
| message | String | 错误消息 | 可以为空 |
| errorCode | int | 结果的代码 | 如果成功，errorCode为0 |
| data | List | 成功后的数据 |  |
| type | int | 令牌类型 | type=1：短信令牌  type=2：硬件令牌  type=4：APP令牌 |
| tokenUsage | TokenUsage | 令牌使用情况 | 引用:6.2.5:TokenUsage |

## 认证系统令牌列表

**URL:** http://ip:port/am/rest/p/1/token/list/active

**请求方式：**GET 或 POST

**参数：**

* **adminLoginName（必填）**管理员登录名
* **adminPassword（必填）**管理员密码
* **tenantName（必填）**站点名称
* **usedInDays(选填，int类型)** 天数

**说明：天数是追溯多少天的数据，如果不选则是系统默认（目前系统默认按最近30天计算）**

**返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 是否成功 | true: 成功  false: 失败 |
| message | String | 错误消息 | 可以为空 |
| errorCode | int | 结果的代码 | 如果成功，errorCode为0 |
| data | list | 返回的结果 |  |
| token | TokenInfo | 令牌信息 | 引用6.2.3:TokenInfo |
| lastUseTime | long(毫秒) | 令牌上次使用时间 |  |

# Portal

## 增量更新用户信息（微信）

**URL:** <http://ip:port/am/rest/p/1/portal/weixin/incSyncUser>

**编码：**所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**参数：**

* **tenantName（必填）**: 商户名称；
* **mobile（必填）**: 用户手机号（会作为登录名）；
* **roleName（可选）**: 角色名；
* **adminLoginName（必填）**: 管理员用户名；
* **adminPassword（必填）**: 管理员密码。

**JSON返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 操作是否成功 | true: 成功  false: 失败 |
| message | string | 错误消息 |  |

**示例：**

<http://192.168.1.30:8080/am/rest/p/1/portal/weixin/incSyncUser?tenantName=TestTenant&loginName=13800000000&mobile=13800000000&roleName=guest&adminLoginName=admin&adminPassword=admin>

示例返回值：

{"success":true,"errorCode":0,"data":null,"message":null}

## 第三方验证码登录验证（微信）

1. 认证方式 —> 微信认证 —> 是否开启，设置为ON，如下图所示；
2. 启用第三方验证码，设置为ON；
3. 第三方微信验证码API URL里面填写第三方验证验证码的接口，如下图所示。
4. 第三方验证码 API RUL 配置项：

地址：HTTP

编码：UTF-8

请求方式：GET

参数：code (必填)验证码

URL示例：http://service.livenaked.com:8080/service-wifi/authAutoCode?code=123456

返回值示例：{"success":true,"errorCode":0,"data":null,"message":null}

若success为false，则表示验证失败。

## 用户登录前获取外部角色（短信，微信）

若获取外部角色开关为开启，用户点击发送短信按钮时，则先通过此配置查找该用户的外部角色，若有角色返回值，则为该用户增加对应角色。

**配置项：**

**外部角色地址：**HTTP或HTTPS URL；

**外部角色秘钥：**验证发送方是否合法。



**参数：**

* **mobile（必传）**: 用户手机号；
* **secretKey（必传）**: 共享秘钥；

**实现此接口的系统需要返回：**

{

"success":true, // 接口调用是否成功，true为允许用户登录，false为禁止用户登录

"data":["role1","role2","role3"], // 用户角色列表，可为空

"message":"成功" // 接口调用消息，用于返回错误消息。可为空

}

返回值为类ResponseData对应的json值，ResponseData 的java代码为：

public class ResponseData{

private boolean success;

private List<String> data;

private String message;

…}

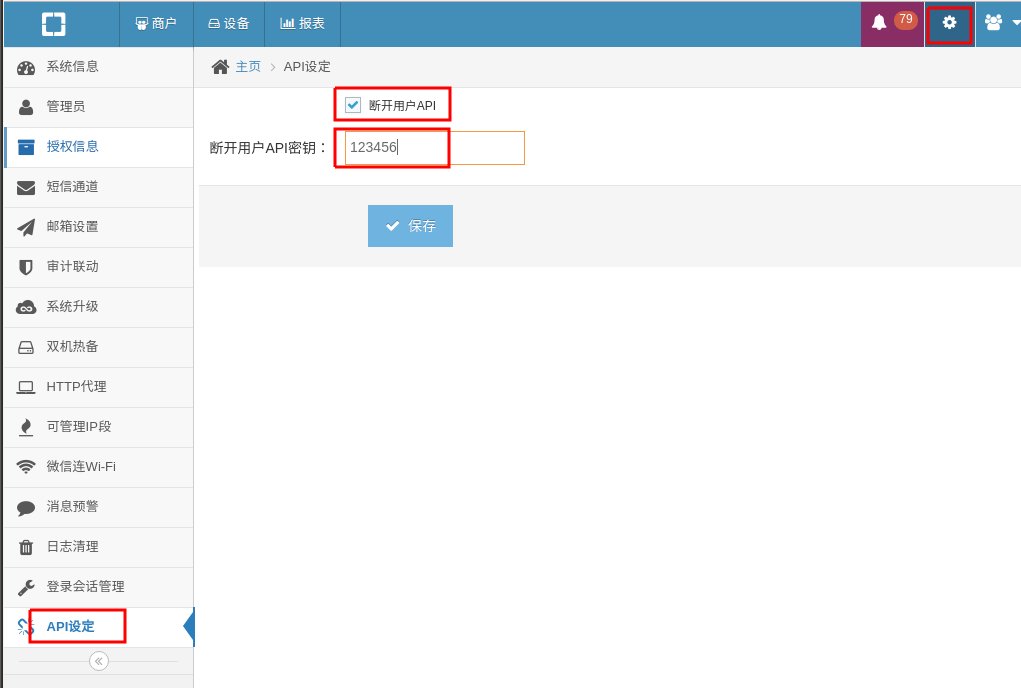
若success为false，则表示该用户无权登录。

## 断开在线用户

**配置项：**

**允许API调用：**系统配置->API设定，开启"断开用户API"；

**API秘钥：**系统配置->API设定，配置"断开用户API密钥"。



1. **通过用户会话ID**

**URL:** <http://ip:port/am/webauth/1/disconnect>

**编码：**所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**参数：**

* **sessionId（必传）**: 用户会话ID；
* **cause（可选）**: 断开用户原因；
* **secretKey（必传）**: API秘钥；
* unbindTerminal（boolean 可选）: 是否解绑终端MAC地址(默认是false，false=不解绑)；

**JSON返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 操作是否成功 | true: 成功  false: 失败 |
| message | string | 错误消息 |  |

**示例：**

<http://192.168.1.30:8080/am/webauth/1/disconnect?sessionId=BD28CKOW&cause=Timeout&secretKey=123456>&unbindTerminal=true

示例返回值：

{"success":true,"errorCode":0,"data":null,"message":null}

1. **通过用户登陆名或终端mac地址**

**URL:** <http://ip:port/am/webauth/1/disconnect>/disconnectByLoginNameOrTerminalMac

**编码：**所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**参数：**

* **tenantName（必传）**: 站点名；
* **loginName（必传）**: 用户登录名;
* **terminalMac (比传)**: 终端mac地址;
* **cause（可选）**: 断开用户原因；
* **secretKey（必传）**: API秘钥；
* unbindTerminal（boolean 可选）: 是否解绑终端MAC地址(默认是false，false=不解绑)；

**注:必传项中loginName与terminalMac,选填其中一个,不要两个都写;**

**JSON返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 操作是否成功 | true: 成功  false: 失败 |
| message | string | 错误消息 |  |

**示例：**

[http://192.168.1.30:8080/am/webauth/1/disconnect/](http://192.168.1.30:8080/am/webauth/1/disconnect?sessionId=BD28CKOW&cause=Timeout&secretKey=123456)disconnectByLoginNameOrTerminalMac?tenantName=zxz&loginName=zzz&cause=Timeout&secretKey=123456&unbindTerminal=true

或

[http://192.168.1.30:8080/am/webauth/1/disconnect/](http://192.168.1.30:8080/am/webauth/1/disconnect?sessionId=BD28CKOW&cause=Timeout&secretKey=123456)disconnectByLoginNameOrTerminalMac?tenantName=zxz&terminalMac=C3:55:33:DD:11:02&cause=Timeout&secretKey=123456&unbindTerminal=true

示例返回值：

{"success":true,"errorCode":0,"data":null,"message":null}

## APP扫码授权（协助扫码）

**例如：电脑连WIFI弹出二维码后，用APP扫描二维码后，电脑可以上网**

**在调用此接口时，确保授权人已经在AM系统中**

**请求方式GET, POST**

**编码：**所有参数的值用URL编码，例如Java：URLEncoder.encode("test", "UTF-8")

**URL:** http(s)://ip:port/am/rest/p/1/portal/qrCode/grant

**参数：**

* **adminLoginName（必填）**: 管理员用户名
* **adminPassword（必填）**: 管理员密码
* **qrCodeSessionId（必填）**: 二维码的会话 （来自二维码）
* **grantorLoginName（必填）**: **登录名 （**授权人）
* **granteeLoginName（必填）**: **登录名 （**被授权人)
* **grantHours（必填）**: 授权小时（单位小时）
* **granteeRoleName（可选）**: 角色名
* **granteePersonalName（可选）**: 姓名（被授权人）
* **granteeMobile（可选）**: 手机号（被授权人）
* **granteeSubject（可选）**: 事由（被授权人）

**JSON返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **定义** | **值** |
| success | boolean | 操作是否成功 | true: 成功  false: 失败 |
| message | string | 错误消息 |  |

成功返回值

{"success":true,"errorCode":0,"data":null,"message":null}

# 附录

## 请求响应码

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **含义** | **说明** |
| success | boolean | 是否成功 | true: 成功  false: 失败 |
| message | String | 错误消息 | 可以为空 |
| errorCode | int | 结果的代码 | 如果成功，errorCode为0 |
| data | List | 返回的结果 |  |

## 数据结构

### TenantInfo

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **含义** | **说明** |
| id | String | 站点ID |  |
| name | String | 站点名称 |  |
| abbreviation | String | 站点名称缩写 |  |
| type | String | 站点类型 |  |
| provinceCode | String | 站点省份 |  |
| cityCode | String | 站点城市 |  |
| areaCode | String | 站点地区 |  |

### StrongAuthSession

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **含义** | **说明** |
| user | UserInfo | 用户信息 | 引用6.2.4:UserInfo |
| passedTokens | List<TokenInfo> | 令牌集合 | 用户可能使用多个令牌登陆;引用6.2.3:TokenInfo |
| accessServerName | String | 设备名 |  |
| loginTime | long(毫秒) | 登录时间 |  |
| loginSuccess | boolean | 登录是否成功 | true:成功;false:失败 |

### TokenInfo

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **含义** | **说明** |
| type | int | 令牌类型 | 1：时间令牌  2：短信令牌 |
| model | int | 令牌模型 | type=1时;  2：硬件令牌  4：APP(手机)令牌  type=2时;  1：短信令牌 |
| serial | String | 令牌序列号 | 令牌序列号(如果是短信令牌，则是手机号) |
| expireTime | long(毫秒) | 令牌到期时间 |  |

### UserInfo

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **含义** | **说明** |
| userId | String | 用户ID |  |
| personalName | String | 真实姓名 |  |
| loginName | String | 登录名 |  |
| mobiles | List<String> | 手机号码 | 用户可能会存储多个手机号 |
| email | String | 邮箱 |  |
| roles | List<String> | 角色 | 用户可能会拥有多个角色 |

### TokenUsage

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **含义** | **说明** |
| usedCount | long(个) | 令牌使用个数 |  |
| unusedCount | long(个) | 令牌未使用个数 |  |

### TokenBinding

|  |  |  |  |
| --- | --- | --- | --- |
| **变量** | **类型** | **含义** | **说明** |
| id | String | 绑定ID |  |
| user | UserInfo | 用户信息 | 引用6.2.4:UserInfo |
| token | TokenInfo | 令牌信息 | 引用6.2.3:TokenInfo |
| priority | int | 优先级 |  |
| bindingEnabled | boolean | 绑定是否生效 | true:生效; false:失效 |
| startTime | long(毫秒) | 绑定生效的开始时间 |  |