# 统一认证鉴权

## 简介

在微服务架构下，一个应用会被拆分成若干个微应用，每个微应用都需要对访问进行鉴权，每个微应用都需要明确当前访问用户以及其权限。尤其当访问来源不只是浏览器，还包括其他服务的调用时，单体应用架构下的鉴权方式就不是特别合适了。在微服务架构下，要考虑外部应用接入的场景、用户–服务的鉴权、服务–服务的鉴权等多种鉴权场景。

比如用户A访问账户系统Account Service，A如果未登录，则首先需要登录，请求获取授权token。获取token之后，A将携带着token去请求访问某个文件，这样就需要对A的身份进行校验，确定A是否可以访问该文件。

为了适应架构的变化、需求的变化，统一认证鉴权模块被单独出来作为一个基础的微服务系统，为其他业务service提供服务。

## 需求

统一认证鉴权模块主要包括两方面需求：

* 用户认证：对于请求的用户身份的认证，保证目前发出请求的用户为合法用户；
* API级别的操作权限控制：这个在第一点之后，当鉴定完用户身份合法之后，对于该用户的某个具体请求是否具有该操作执行权限进行校验。

### 用户认证

用户认证的功能主要有：登录，注销，刷新token以及修改密码。

**登录：**

物联网云平台为每位用户配置了一套独特的用户名和密码，用户可以使用各自的这套用户名和密码来使用系统，以便系统能识别该用户的身份，从而保持该用户的使用习惯或使用数据。

用户输入自己的账号与密码，这些信息经由API网关转发到统一认证鉴权模块，认证鉴权模块从数据库中获取用户密码并且进行比对，如果匹配正确，则用户认证通过，否则返回错误信息。

|  |  |  |
| --- | --- | --- |
| 执行操作 | 传入参数 | 返回结果 |
| 输入账号和密码，调用相关接口，进行登录操作 | 用户输入的账号与密码 | 如果登陆成功，返回用户所需令牌。否则返回错误信息。 |

**注销：**

向物联网云平台发出清除现在登录的用户的请求，清除后即可使用其他用户来登录平台。

|  |  |  |
| --- | --- | --- |
| 执行操作 | 传入参数 | 返回结果 |
| 用户调用相关接口，进行注销操作 | N/A | 注销成功，返回“注销成功”。否则返回“注销失败” |

**刷新token：**

由于token的时效一般不会很长，而refresh token一般周期会很长，为了不影响用户的体验，可以使用refresh token去动态的刷新token。

|  |  |  |
| --- | --- | --- |
| 执行操作 | 传入参数 | 返回结果 |
| 调用相关接口，刷新用户的access\_token | 用户在登录成功后，获取到的fresh\_token | 如果刷新成功，返回新的令牌。否则返回错误信息。 |

**修改密码：**

已经经过身份认证的用户，可以调用相关接口，并且提供当前用户的密码作为认证，就可以对当前密码进行修改操作。

|  |  |  |
| --- | --- | --- |
| 执行操作 | 传入参数 | 返回结果 |
| 调用相关接口，修改用户的密码 | 用户修改前的密码与用户的新密码 | 如果旧密码验证成功，则密码修改成功，否则修改失败返回错误信息 |

**相关接口：**

auth/login（登录）

* 描述：进行登录操作
* 请求类型：POST
* 参数：

| **#** | **name** | **desc** | **是否必要** |
| --- | --- | --- | --- |
| 1 | username | 用户名 | 是 |
| 2 | password | 密码 | 是 |

* 返回示例：

{

"access\_token": " access token value ",

"token\_type": "bearer",

"refresh\_token": " refresh token value ",

"expires\_in": 43195,

"scope": "all",

}

auth/logout（注销用户）

* 描述：进行登出操作
* 请求类型：GET
* 参数：N/A
* 返回：注销成功则返回“注销成功”，否则返回“注销失败”

auth/refresh\_token（刷新token）

* 描述：刷新access token
* 请求类型：GET
* 参数：

| **#** | **name** | **desc** | **是否必要** |
| --- | --- | --- | --- |
| 1 | refresh\_token | 此前获得的refresh\_token | 是 |

* 返回示例：

{

"access\_token": " access token value ",

"token\_type": "bearer",

"refresh\_token": " refresh token value ",

"expires\_in": 43195,

"scope": "all",

}

auth/changePassword（修改密码）

* 描述：进行密码修改操作
* 请求类型：POST
* 参数：

| **#** | **name** | **desc** | **是否必要** |
| --- | --- | --- | --- |
| 1 | currentPassword | 用户当前密码 | 是 |
| 2 | newPassword | 用户的新密码 | 是 |

* 返回：状态码200

### API级别权限控制

当鉴定完用户身份合法之后，对于该认证用户，我们还需要对他所进行的操作进行鉴权，以确定该用户是否拥有执行此项操作的权力。例如，一个Customer\_User权限的用户，如果想要执行删除设备的操作，就需要通过统一认证鉴权模块进行鉴权，该模块进行权力鉴定并且发现该用户的权限为Customer\_User，而删除设备的API调用操作只有Tenant\_Admin权限的用户才能够执行，此时Customer\_User的API调用请求将被禁止，并且返回错误信息。

每当用户发出请求时，请求头部都会携带一个Token，该Token会被转发到统一认证鉴权模块进行合法性验证（确定该Token是系统签发）以及鉴权（确定该用户的权限），并将结果返回。各个微服务会根据返回结果来确定该请求的权限是否符合要求。符合要求则执行该服务，否则拒绝执行。

|  |  |  |
| --- | --- | --- |
| 执行操作 | 传入参数 | 返回结果 |
| 其他服务调用鉴权接口，进行权限鉴定 | 请求中携带的access\_token | token合法，返回鉴权结果，否则报错。 |

**鉴权**的API接口为：

auth/check\_token（鉴权）

* 描述：进行鉴权，确定该Token的合法性和权限
* 请求类型：POST
* 参数：

| **#** | **name** | **desc** | **是否必要** |
| --- | --- | --- | --- |
| 1 | token | 此前获得的access\_token | 是 |

* 返回示例：

{

"UserId": "d6448c24-3c4c-4b80-8372-c2d61868f8c6",

"user\_name": "Tenant",

"scope": [

"all"

],

}

## 各权限所能调用的接口

### 管理员（SYS\_ADMIN）：

系统管理员的权限主要包括对系统设置的操作上。例如可以对系统插件、系统规则进行管理。同时，系统管理员还拥有增加租户管理员（TENANT\_ADMIN）以及对租户管理员进行管理的权限。他人向系统管理员申请平台的使用权，系统管理员为其添加新的租户与租户管理员账号。但是，系统管理员无权干涉租户内部的事物，包括设备、设备组的管理，租户自定义的插件、规则、服务管理，以及租户内的客户（Customer）管理。

以下是系统管理员能够调用的接口清单：

**PLUGIN：**

getPluginById

getPluginByToken

savePlugin

activatePluginById

suspendPluginById

getSystemPlugins

getTenantPlugins

deletePlugin

**RULE**

getRuleById

getRulesByPluginToken

saveRule

activateRuleById

suspendRuleById

getSystemRules

getTenantRules

deleteRule

**TENANT**

getTenantById

saveTenant

deleteTenant

getTenants

getTenantAdmins

**USER**

getUserById

saveUser

deleteUser

getTenantAdmins

### 租户管理员（TENANT\_ADMIN）：

租户管理员的权限主要包括对租户内部（包括设备、设备组、服务组、租户插件、租户规则以及客户Customer）的管理。由于设备是属于租户的资产，因此租户管理员对设备拥有最高的权限。

以下是租户管理员能够调用的接口清单：

**CUSTOMER：**

getCustomerById

saveCustomer

deleteCustomer

getCustomers

**DEVICE:**

getDeviceById

saveDevice

updateDeviceCoordinate

deleteDevice

assignDeviceToCustomer

unassignDeviceFromCustomer

assignDeviceToPublicCustomer

getDeviceCredentialsByDeviceId

saveDeviceCredentials

getTenantDevices

getTenantDevice

getCustomerDevices

getDevicesByIds

getDeviceTypes

**GROUP:**

ceateGroup

deleteGroup

listGroups

assignDeviceToGroup

unAssignDeviceFromGroup

getDevicesByGroupId

**PLUGIN:**

getPluginById

getPluginByToken

savePlugin

activatePluginById

suspendPluginById

getTenantPlugins

deletePlugin

**RULE:**

getRuleById

getRulesByPluginToken

saveRule

activateRuleById

suspendRuleById

getTenantRules

deleteRule

**SERVICE:**

findAllManufactures

findDeviceTypesByManufacture

findModelsByManufactureAndDeviceType

save

delete

addServiceToGroup

deleteServiceFromServiceGroup

findServiceTables

**TENANT:**

getTenantById

**USER:**

getUserById

saveUser

deleteUser

getCustomerUsers

### 客户用户（CUSTOMER\_USER）：

客户用户的权限比租户管理员的权限更低，只能对设备进行查看和信息的修改，无法对设备进行删除。此外，客户用户能够对设备进行分类，如果设备组由该用户创建，那么该用户就对该设备组拥有最高的权限。

以下是客户用户能够调用的接口清单：

**DEVICE：**

getDeviceById

saveDevice

updateDeviceCoordinate

getDeviceCredentialsByDeviceId

getCustomerDevices

getDevicesByIds

getDeviceTypes

**GROUP**：

ceateGroup

deleteGroup

listGroups

assignDeviceToGroup

unAssignDeviceFromGroup

getDevicesByGroupId

**USER：**

getUserById

saveUser