

用户授权服务设计文档

用户授权服务设计文档

- 需求
- 总体设计
 - 项目架构
- 名词定义
- 事件风暴
 - 管理接口 sa
 - 开放接口 poa
 - 同步引擎 sync-engine
- 领域设计
 - 管理接口 sa
 - 实体 Entity
 - 应用
 - 角色
 - 角色组
 - 授权
 - 分级授权
 - 授权日志
 - Value Object
 - 应用
 - 角色
 - 角色组
 - 授权
 - 分级授权
 - 授权日志
 - 事件 Event
 - 业务 Service
 - 开放接口 poa
 - 实体 Entity
 - 用户角色
 - Value Object
 - 用户角色
- 项目设计
 - 项目目录
 - 公用项目
 - 管理接口 sa
 - 开放接口 poa
- 功能设计
 - 管理接口 sa
 - 功能说明
 - 应用
 - 角色
 - 角色组
 - 授权批次

授权
分级授权
授权日志
授权统计
接口定义
开放接口 poa
功能说明
用户角色
接口定义

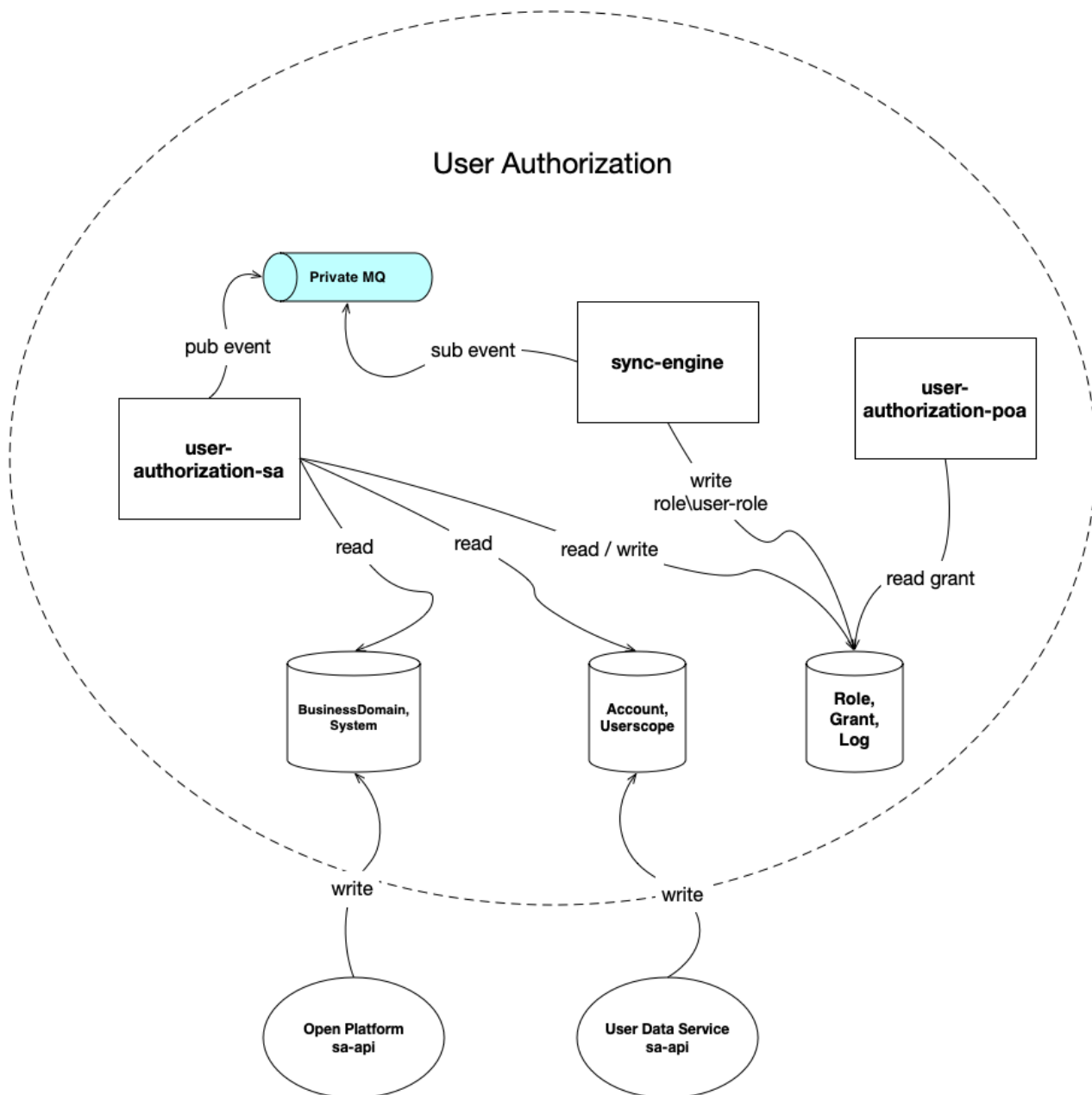
需求

参考：

- 需求文档：授权管理-需求文档-v1.1_19-07-08.docx <https://supwisdom.coding.net/p/UM/attachment/1066238/preview/1066574>
- 产品原型：授权服务-产品原型-v2.0_19-07-08.rp <https://supwisdom.coding.net/p/UM/attachment/1066238/preview/1066579>

总体设计

项目架构

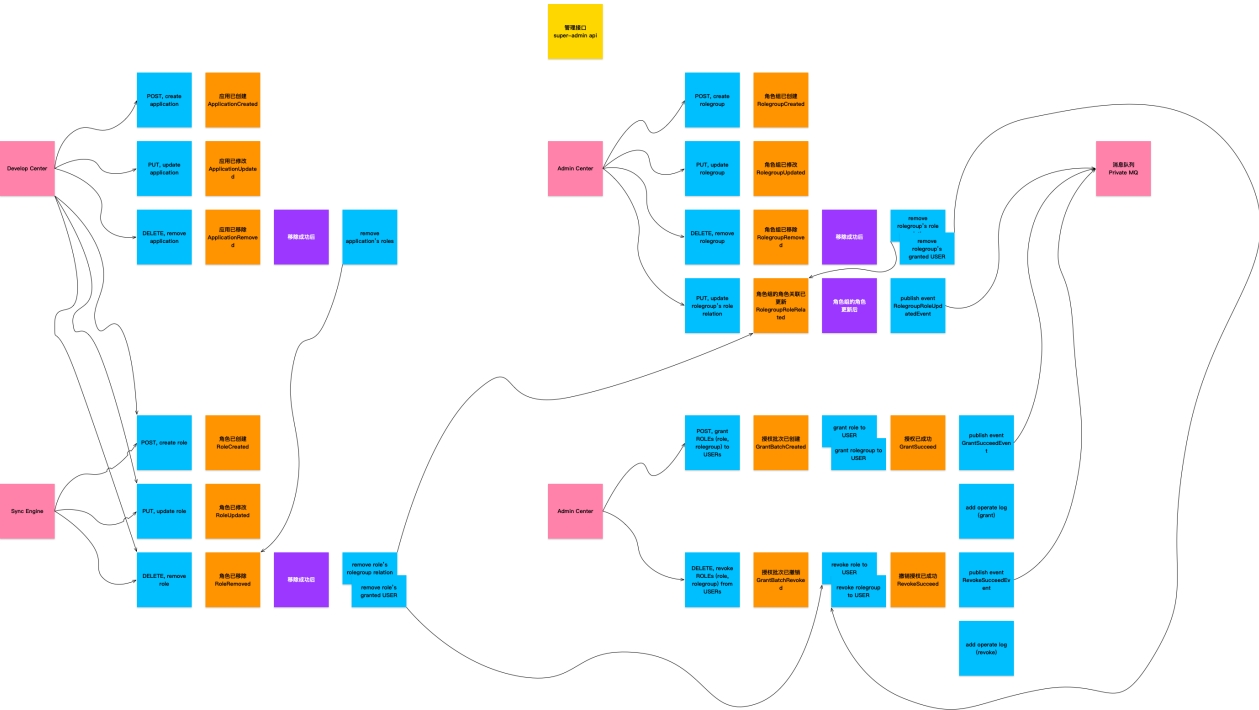


名词定义

- 用户，此文档中，用户是一种泛指，即一个范围，可以是一个帐号、可以是一个用户规则
- 帐号，单个人的某一个帐号
- 用户规则，符合相关规则的一批用户的帐号
- 应用，指第三方对接应用，记录其应用标识 applicationId、角色同步接口地址 syncUrl 等信息
- 角色，此文档中，角色有时指一个角色，有时泛指角色和角色组
- 角色组，是一组角色的集合，可随意定义
- 权限，泛指，菜单权限、操作权限
- 授权，也称为 用户授权，即将角色、角色组 分配到 用户（帐号、用户规则），使用户在相关应用下具有角色上相应的权限
- 分级授权，将 授权、分级授权 这类操作 分配到某个帐号，使这个帐号在授权管理下具有相关的操作权限
- 可授权，授权管理下一种操作权限，表示可以进行 授权（用户授权） 操作
- 可管理，授权管理下一种操作权限，表示可以进行 分级授权 操作

事件风暴

管理接口 sa



包括：

- 应用管理

对应用的信息进行维护

场景一：

1. 开发者，在开放平台，授权服务基础能力中，进行申请，提交 应用的同步角色接口 (syncUrl)
2. 开放平台，调用 应用管理接口，创建应用，并获得应用标识 (applicationId)

场景二：

1. 开发者，在开放平台，授权服务基础能力中，修改 应用的同步角色接口 (syncUrl)
2. 开放平台，调用 应用管理接口，修改应用

- 角色管理

对应用的角色进行维护

场景一：

1. 同步引擎，根据各个应用的同步角色接口 (syncUrl) ，抓取到应用下的角色列表
2. 同步引擎，调用 角色管理接口，创建 或 修改 角色

场景二：

1. 开发者，在开放平台，授权服务基础能力中，创建 或 修改 角色
2. 开放平台，调用 角色管理接口，创建 或 修改 角色

- 角色组管理

对角色组进行管理

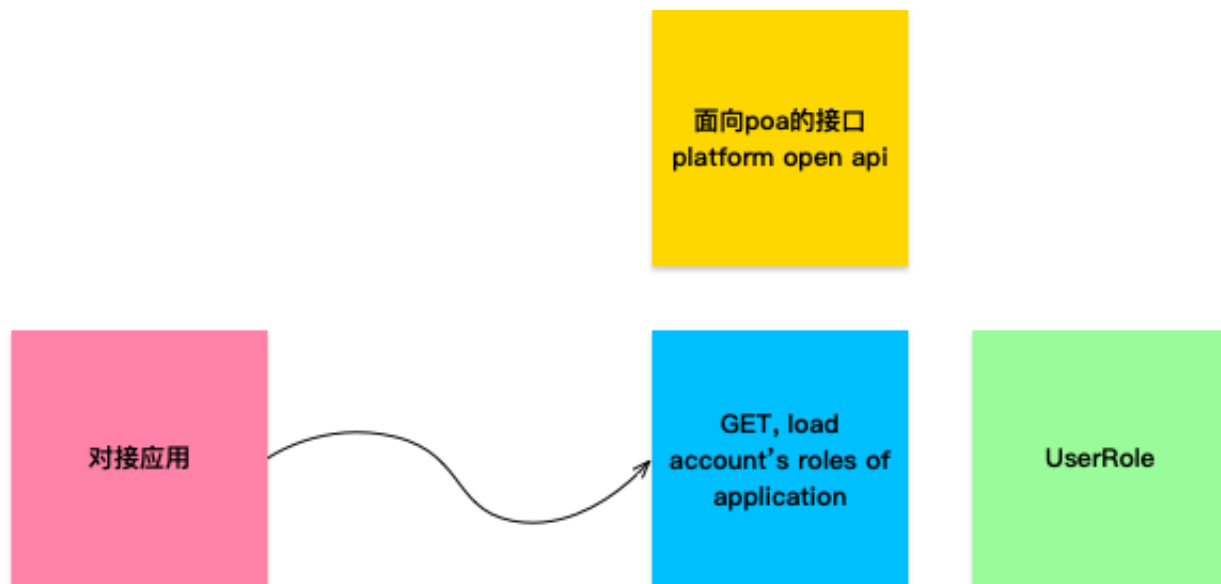
场景一：

1. 管理员，在管理中心，对角色组进行 创建、修改、删除 或 关联角色等操作

- 授权管理

授权操作：用户授权，用户规则授权

开放接口 poa



包括：

- 获取用户角色

第三方应用，根据应用标识、用户名，获取到该用户在其应用中所拥有的角色列表

场景：

1. 第三方应用，须将授权管理功能托管到用户授权管理
2. 第三方应用（或其开发者），在开放平台，授权服务基础能力中，申请使用能力，取得 applicationId
3. 第三方应用（或其开发者），在开放平台，授权服务基础能力中，提交 应用的同步角色接口 (syncUrl)，或 直接创建角色
4. 第三方应用（或其开发者），在开放平台，授权服务基础能力中，对角色进行授权，或 管理员 在管理中心，授权管理中，进行授权操作
5. 第三方应用（或其开发者），在开放平台，平台Open API 基础能力中，申请使用能力，取得 clientId、clientSecret
6. 第三方应用（或其开发者），在开放平台，平台Open API 基础能力中，申请服务 Scope (userAuthorizationServicePoa:v1:readUserRole)
7. 第三方应用（或其开发者），依照 [应用对接指南 - 示例](#)，进行角色获取的对接开发，调用接口 /apis/userAuthorizationServicePoa/v1/roles/userRoles，
8. 第三方应用（或其开发者），根据获取的当前用户的角色，判定该用户在应用中的权限范围

同步引擎 sync-engine

包括：

- 定时同步应用的角色

根据应用的同步角色接口（syncUrl），定时获取第三方应用的角色列表（间隔 5 分钟 或 基于配置）

- 定时更新用户角色的关系

由于 用户规则 下的人员会不定期地变更，而且数据庞大，所以，需要一种机制，保证 用户规则 对应的角色、角色组，能够准实时地更新到 用户规则下的帐号

基于各个用户规则，采用定时线程的方式（间隔 5 分钟 或 基于配置），获取到 用户规则下的帐号列表，并根据 用户规则对应的角色、角色组，更新用户角色关系

- 异步拆分用户角色的关系

为了方便查询用户的角色，当 管理员完成 用户授权 操作后，需要将授权信息进行拆分，即拆分为 帐号 和 角色 的直接关系，并需要在 角色被移除、角色组被移除、角色组下的角色关联被更新 时，须同时 更新用户角色关系

实现方式：

1. 管理接口 sa 服务中，角色被移除、角色组被移除、角色组下的角色关联被更新 时，发布事件
2. 同步引擎 sync-engine 中，订阅事件，一旦监听到事件，则 更新用户角色关系

领域设计

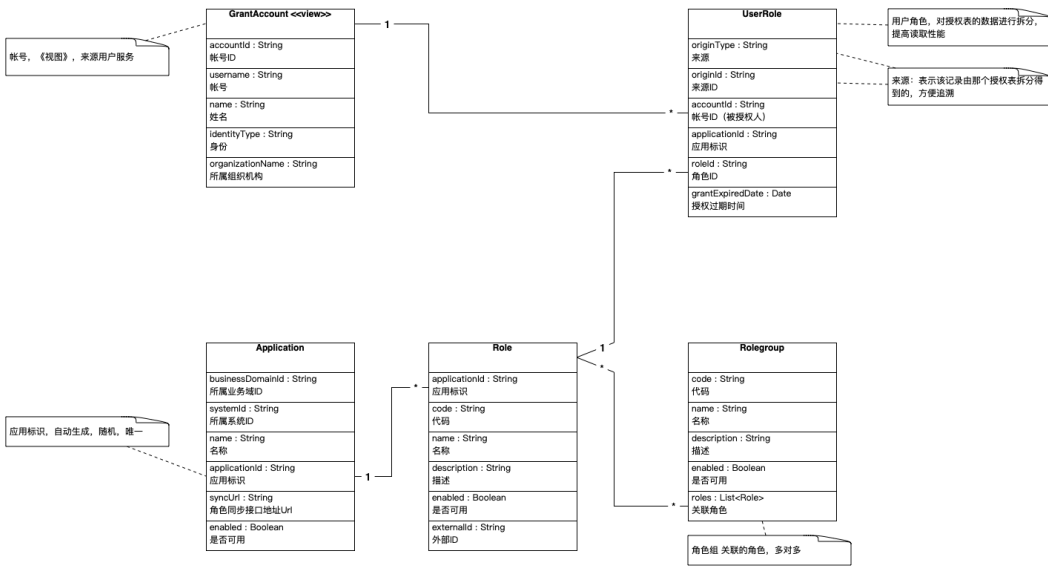
管理接口 sa

	Role context			Rolegroup context		
Entities	<div>Application</div> <div>Role</div>	<div>ApplicationRepository</div> <div>RoleRepository</div>		<div>Rolegroup</div>	<div>RolegroupRepository</div>	
Value Objects	<div>ApplicationCreateRequest</div> <div>RoleCreateRequest</div>	<div>ApplicationUpdateRequest</div> <div>RoleUpdateRequest</div>		<div>RolegroupCreateRequest</div> <div>RolegroupRelateRoleRequest</div>	<div>RolegroupUpdateRequest</div>	
Events	<div>ApplicationCreated</div> <div>RoleCreated</div>	<div>ApplicationUpdated</div> <div>RoleUpdated</div>	<div>ApplicationRemoved</div> <div>RoleRemoved</div>	<div>RolegroupCreated</div> <div>RolegroupRoleRelated</div>	<div>RolegroupUpdated</div>	<div>RolegroupRemoved</div>
Services	<div>AutoIdGenerater</div> <div>RoleCodeExistValidator</div>	<div>ApplicationIdValidator</div>	<div>SyncUrlValidator</div>	<div>RolegroupCodeExistValidator</div>		
DTOs	<div>ApplicationCreateResponse</div> <div>RoleCreateResponse</div>	<div>ApplicationUpdateResponse</div> <div>RoleUpdateResponse</div>	<div>ApplicationRemoveResponse</div> <div>RoleRemoveResponse</div>	<div>RolegroupCreateResponse</div> <div>RolegroupRelateRoleResponse</div>	<div>RolegroupUpdateResponse</div>	<div>RolegroupRemoveResponse</div>

	Grant context		Data Grant context		Grant Log context	
Entities	<div>GrantedAccountRole</div> <div>GrantedAccountRolegroup</div> <div>GrantedUserScopeRole</div> <div>GrantedUserScopeRolegroup</div>	<div>GrantedAccountRoleRepository</div> <div>GrantedAccountRolegroupRepository</div> <div>GrantedUserScopeRoleRepository</div> <div>GrantedUserScopeRolegroupRepository</div>	<div>DataGrantedAccountRole</div>	<div>DataGrantedAccountRoleRepository</div>	<div>GrantOperateLog</div> <div>GrantAccessLog</div>	<div>GrantOperateLogRepository</div> <div>GrantAccessLogRepository</div>
Value Objects	<div>GrantedAccountRolesPostRequest</div> <div>GrantedUserscopeRolesPostRequest</div>	<div>GrantedRoleAccountsPostRequest</div>	<div>DataGrantedAccountRoleCreateRequest</div>	<div>DataGrantedAccountRoleRevokeRequest</div>		
Events	<div>GrantBatchCreated</div> <div>GrantSucceed</div>	<div>GrantBatchRevoked</div> <div>RevokeSucceed</div>				
Services						
DTOs	<div>GrantedAccountRolesPostResponse</div> <div>GrantedUserscopeRolesPostResponse</div>	<div>GrantedRoleAccountsPostResponse</div>	<div>DataGrantedAccountRoleCreateResponse</div>	<div>DataGrantedAccountRoleRevokeResponse</div>		

GrantBatch	GrantBatchRepository
------------	----------------------

实体 Entity

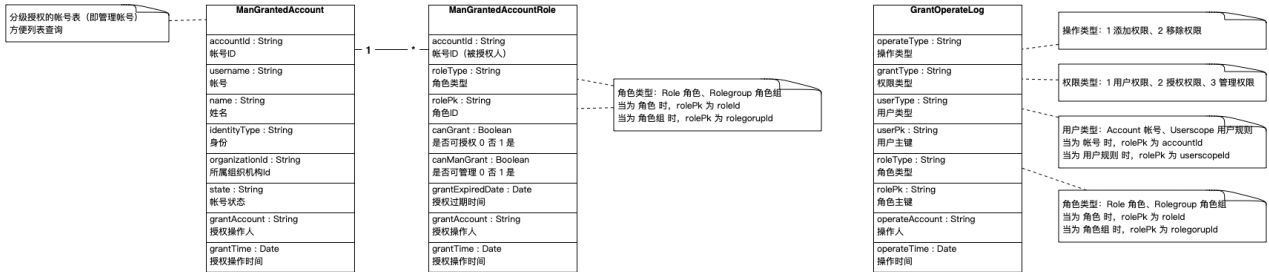


GrantedAccountRole
accountid: String 帐号ID (被授权人)
roleId: String 角色ID
grantExpiredDate: Date 授权过期时间

GrantedUserscopeRole
userscopeId: String 用户规则ID (被授权人)
roleId: String 角色ID
grantExpiredDate: Date 授权过期时间

GrantedAccountRolegroup
accountid: String 帐号ID (被授权人)
rolegroupId: String 角色组ID
grantExpiredDate: Date 授权过期时间

GrantedUserscopeRolegroup
userscopeId: String 用户规则ID (被授权人)
rolegroupId: String 角色组ID
grantExpiredDate: Date 授权过期时间



GrantBatch
batchNo: String 批次号
batchStatus: String 批次状态 (1 生效 Grant, 2 撤销 Cancel)
grantedUserSummary: String 授予用户摘要
grantedRoleSummary: String 授予角色摘要
grantExpiredDate: Date 授权过期时间
grantAccount: String 授权操作人
grantTime: Date 授权操作时间
cancelAccount: String 撤销授权操作人
cancelTime: Date 撤销授权操作时间

用户指: 帐号、用户规则 摘要, 简单说明即可
角色指: 角色、角色组 摘要, 简单说明即可

GrantBatchDetail
batchId: String 批次ID
operateType: String 操作类型
userType: String 用户类型
userPk: String 用户主键
roleType: String 角色类型
rolePk: String 角色主键
grantAccount: String 授权操作人
grantTime: Date 授权操作时间
cancelAccount: String 撤销授权操作人
cancelTime: Date 撤销授权操作时间

操作类型: 1 添加权限、2 移除权限
用户类型: Account 帐号、Userscope 用户规则 当为 帐号 时, rolePk 为 accountid 当为 用户规则 时, rolePk 为 userscopeId
角色类型: Role 角色、Rolegroup 角色组 当为 角色 时, rolePk 为 roleId 当为 角色组 时, rolePk 为 rolegroupid

应用

- Application

应用表

应用标识 applicationId, 由服务自动随机生成, 保证唯一性

角色

- Role

角色表

应用标识 applicationId，匹配 Application 的 applicationId

外部ID externalId，存放角色在源应用的角色表中的ID，便于数据的更新

角色组

- Rolegroup

角色组表

关联角色 roles，与 Role 形成多对多的关系

授权

- GrantBatch

授权批次表

批次号 batchNo，根据时间戳自动生成即可，保证唯一，yyyyMMddHHmmss

授予用户摘要 grantedUserSummary，简单的描述信息。用户是广义的，帐号、用户规则的泛指

授予角色摘要 grantedRoleSummary，简单的描述信息，角色是广义的，角色、角色组的泛指

授权过期时间 grantExpiredDate，若为空，则该授权长期有效；否则，到了过期时间，该授权自动失效

授权操作人 grantAccount，此次授权操作的帐号，方便追溯、替换

- GrantedAccountRole

已授权的帐号 - 角色 表

- GrantedUserscopeRole

已授权的用户规则 - 角色 表

- GrantedAccountRolegroup

已授权的帐号 - 角色组 表

- GrantedUserscopeRolegroup

已授权的用户规则 - 角色组 表

分级授权

- DataGrantedAccountRole

角色操作的分级授权表，即指定 角色、角色组数据 由谁授权、由谁 分级授权

角色类型 roleType，Role 角色、Rolegroup 角色组，当为 角色 时，rolePk 为 roleId；当为 角色组 时，rolePk 为 rolegroupId

授权过期时间 grantExpiredDate，若为空，则该授权长期有效；否则，到了过期时间，该授权自动失效

授权操作人 grantAccount，此次授权操作的帐号，方便追溯、替换

授权日志

- GrantOperateLog

授权操作日志表

- GrantAccessLog

授权访问日志表

Value Object

应用

- ApplicationQueryRequest
请求对象，分页查询应用
- ApplicationCreateRequest
请求对象，创建应用
- ApplicationUpdateRequest
请求对象，修改应用
- ApplicationQueryResponse
响应对象，分页查询应用
- ApplicationLoadResponse
响应对象，根据 id 获取应用
- ApplicationCreateResponse
响应对象，创建应用
- ApplicationUpdateResponse
响应对象，修改应用
- ApplicationDeleteResponse
响应对象，根据 id 删除应用

角色

- RoleQueryRequest
请求对象，分页查询角色
- RoleCreateRequest
请求对象，创建角色
- RoleUpdateRequest
请求对象，修改角色
- RoleQueryResponse
响应对象，分页查询角色
- RoleLoadResponse
响应对象，根据 id 获取角色
- RoleCreateResponse
响应对象，创建角色

- RoleUpdateResponse
响应对象，修改角色
- RoleDeleteResponse
响应对象，根据 id 删除角色

角色组

- RolegroupQueryRequest
请求对象，分页查询角色组
- RolegroupCreateRequest
请求对象，创建角色组
- RolegroupUpdateRequest
请求对象，修改角色组
- RolegroupQueryResponse
响应对象，分页查询角色组
- RolegroupLoadResponse
响应对象，根据 id 获取角色组
- RolegroupCreateResponse
响应对象，创建角色组
- RolegroupUpdateResponse
响应对象，修改角色组
- RolegroupDeleteResponse
响应对象，根据 id 删除角色组
- RolegroupRelateRoleRequest
请求对象，角色组关联角色
- RolegroupRelateRoleResponse
响应对象，角色组关联角色

授权

分级授权

授权日志

事件 Event

- RolegroupRoleUpdatedEvent

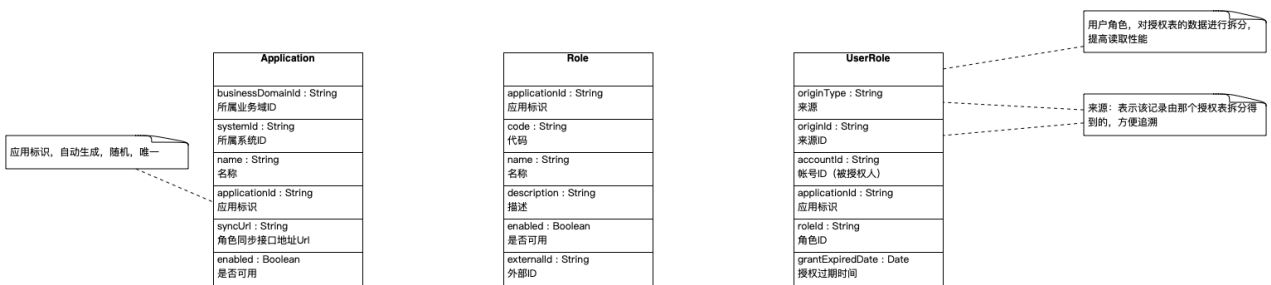
- GrantSucceedEvent
- RevokeSucceedEvent

业务 Service

开放接口 poa

	Role context	
Entities	Role	RoleRepository
	UserRole	UserRoleRepository
Value Objects	UserRolesLoadRequest	RoleUsersLoadRequest
Events		
Services		
DTOs	UserRolesLoadResponse	RoleUsersLoadResponse

实体 Entity

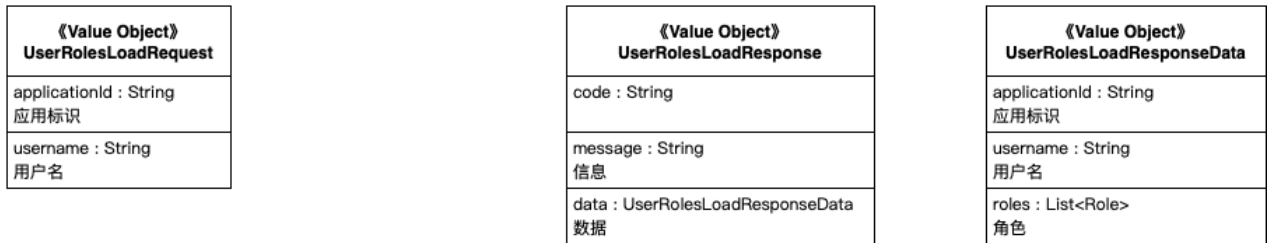


用户角色

- UserRole

用户角色表

Value Object



用户角色

- UserRolesLoadRequest
请求对象，获取用户的角色
- UserRolesLoadResponse
响应对象，获取用户的角色

项目设计

项目目录

目录名	说明
api-docs	接口设计文档
doc	项目文档
doc/requirement	需求
doc/design	设计
doc/installation	安装
sql	数据库建表脚本、初始化数据脚本
deploy-manifests	部署文档
deploy-manifests/k8s	k8s 相关部署文档
user-authorization-common	公用项目
user-authorization-sa	管理接口项目
user-authorization-poa	平台开放接口项目

公用项目

项目目录 user-authorization-common

项目包命名

com.supwisdom.institute.common	公共基类（如，entity 基类、vo 基类 等）
com.supwisdom.institute.user.authorization.service.common	项目公共类

管理接口 sa

项目目录 user-authorization-sa

项目包命名

com.supwisdom.institute.user.authorization.service.sa	
configuration	项目配置类
listener	监听器
utils	工具类
application	应用（业务）
entity	实体类，须继承

repository	存储类
service	业务类（@Service 注解，调用 remote 获取到 JSONObject 对象后，转换成 JOPO 对象，然后由 Controller 使用）
vo	值对象（如，请求、响应对象）
webapi.admin	接口（@RestController 注解，调用 service 获取到 JOPO 对象后，转换成 vo，返回响应）
role	角色（业务）
rolegroup	角色组（业务）
grantbatch	授权批次（业务）
granted	用户授权（业务）
datagranted	分级授权（业务）
grantlog	授权日志（业务）

开放接口 poa

项目目录 user-authorization-poa

项目包命名

com.supwisdom.institute.user.authorization.service.poa	
configuration	项目配置类
listener	监听器
utils	工具类
role	角色（业务）
entity	实体类，须继承
repository	存储类
service	业务类（@Service 注解，调用 remote 获取到 JSONObject 对象后，转换成 JOPO 对象，然后由 Controller 使用）

vo 值对象（如，请求、响应对象）

webapi 接口（@RestController 注解，调用 service 获取到 JOPO 对象后，转换成 vo，返回响应）

功能设计

管理接口 sa

功能说明

应用

- 分页查询应用

```
GET /v1/admin/applications
```

```
pageIndex=0&pageSize=20&mapBean[businessDomainId]=1&mapBean[systemId]=1&mapBean[applicationId]=1&mapBean[name]=示例应用&mapBean[enabled]=true
```

- 根据 id 获取应用详情

```
GET /v1/admin/applications/{id}
```

- 创建应用

```
POST /v1/admin/applications
```

```
{
  "businessDomainId": "1",
  "systemId": "1",
  "name": "示例应用",
  "syncUrl": "https://example.com/api/v1/roles",
  "enabled": true
}
```

响应：


```
{
  "code": 0,
  "message": null,
  "data": {
    "id": "1",
    "businessDomainId": "1",
    "systemId": "1",
    "name": "示例应用",
    "applicationId": "1",
    "syncUrl": "https://example.com/api/v1/roles",
    "enabled": true
  }
}
```

- 修改应用

```
PUT /v1/admin/applications/{id}

{
  "businessDomainId": "1",
  "systemId": "1",
  "name": "示例应用2",
  "syncUrl": "https://example.com/api/v1/roles",
  "enabled": true
}
```

响应:

```
{
  "code": 0,
  "message": null,
  "data": {
    "id": "1",
    "businessDomainId": "1",
    "systemId": "1",
    "name": "示例应用2",
    "applicationId": "1",
    "syncUrl": "https://example.com/api/v1/roles",
    "enabled": true
  }
}
```

- 根据 id 删除应用

删除为物理删除。删除时，同时删除应用下的角色。

```
DELETE /v1/admin/applications/{id}
```

- 根据应用标识 applicationId 获取应用

```
GET /v1/admin/applications/applicationId/{applicationId}
```

响应:

```
{
  "code": 0,
  "message": null,
  "data": {
    "id": "1",
    "businessDomainId": "1",
    "systemId": "1",
    "name": "示例应用",
    "applicationId": "1",
    "syncUrl": "https://example.com/api/v1/roles",
    "enabled": true
  }
}
```

角色

- 分页查询角色

```
GET /v1/admin/roles
```

```
pageIndex=0&pageSize=20&mapBean[applicationId]=1&mapBean[code]=teacher&mapBean[name]=教师&mapBean[enabled]=true
```

- 根据 id 获取角色详情

```
GET /v1/admin/roles/{id}
```

- 创建角色

```
POST /v1/admin/roles
```

- 修改角色

```
PUT /v1/admin/roles/{id}
```

- 根据 id 删除角色

删除为物理删除。删除时，同时删除与角色组的关系（RolegroupRole）；

撤销与人员帐号的关系（GrantedAccountRole）；

撤销与用户规则的关系（GrantedUserscopeRole）；

撤销，仅修改授权状态 为 已撤销，并记录撤销时间。记录授权日志：撤销，角色被删除

```
DELETE /v1/admin/roles/{id}
```

- 获取应用下的角色

请求传入 应用标识 (applicationId)

```
GET /v1/admin/roles/applicationId/{applicationId}
```

角色组

- 分页查询角色组

```
GET /v1/admin/rolegroups
```

```
pageIndex=0&pageSize=20&mapBean[code]=teacher&mapBean[name]=教师  
&mapBean[enabled]=true
```

- 根据 id 获取角色组详情

```
GET /v1/admin/rolegroups/{id}
```

- 创建角色组

```
POST /v1/admin/rolegroups
```

- 修改角色组

```
PUT /v1/admin/rolegroups/{id}
```

- 根据 id 删除角色组

```
DELETE /v1/admin/rolegroups/{id}
```

删除为物理删除。删除时，同时删除与角色的关系 (RolegroupRole)；撤销与人员帐号的关系 (GrantedAccountRolegroup)；撤销与用户规则的关系 (GrantedUserscopeRolegroup)；

撤销，仅修改授权状态 为 已撤销，并记录撤销时间

- 获取角色组下的角色

```
GET /v1/admin/rolegroups/{id}/roles

loadAll=false&pageIndex=0&pageSize=20
```

- 关联角色

```
POST /v1/admin/rolegroups/{id}/roles

{
  "addRoleIds": ["1", "2"],
  "delRoleIds": ["3", "4"],
}
```

授权批次

- 分页查询授权批次

```
GET /v1/admin/grantBatches

pageIndex=0&pageSize=20&operateAccount=1&mapBean[batchStatus]=1&mapBean[grantTimeBegin]=2019-09-22&mapBean[grantTimeEnd]=2019-09-25
```

- 根据 id 获取授权批次详情

```
GET /v1/admin/grantBatches/{id}
```

- 创建授权批次

```
POST /v1/admin/grantBatches
```

- 修改授权批次

```
PUT /v1/admin/grantBatches/{id}
```

- 撤销授权批次

```
GET /v1/admin/grantBatches/{id}/cancel

operateAccount=1
```

请求传入 批次ID

根据批次对应的 GrantedAccountRole, GrantedUserscopeRole, GrantedAccountRolegroup, GrantedUserscopeRolegroup 记录, 撤销授权

撤销授权, 仅修改授权状态 为 已撤销, 并记录撤销时间; 已撤销的记录不再更新撤销状态、撤销时间;

授权

- 添加授权，按人员授权，获取 已选人员 共有的 角色/组

```
GET /v1/admin/granted/grantedAccountRoles
```

```
operateAccount=1&accountIds=1,2,3
```

请求传入 人员列表 (accountIds) ， 查询到所有人员 共同拥有的 角色 (roleIds) 或 角色组 (rolegroupIds) ， 返回

- 添加授权，按人员授权，提交 已选人员，待添加的 角色/组，待移除的 角色/组

```
POST /v1/admin/granted/grantedAccountRoles
```

```
{
  "operateAccount": "1",
  "grantExpiredDate": "2019-07-23 17:48:00",
  "accountIds": ["1","2","3"],
  "addRoleIds": ["1","2"],
  "addRolegroupIds": ["3"],
  "delRoleIds": ["9"],
  "delRolegroupIds": ["4","5"]
}
```

请求传入 已选的人员 (accountIds) ， 待添加的角色 (addRoleIds) 或 角色组 (addRolegroupIds) ， 待移除的角色 (delRoleIds) 或 角色组 (delRolegroupIds) （一般为共同拥有的）

更新 人员 与 角色 或 角色组 的授权关系

- 添加授权，按角色/组授权，获取 已选角色/组 共有的 人员

```
GET /v1/admin/granted/grantedRoleAccounts
```

```
operateAccount=1&roleIds=1,2,3&rolegroupIds=4,5
```

请求传入 已选的角色 (roleIds) /角色组 (rolegroupIds) ， 查询到 角色/组 共同拥有的人员 (accountIds)

- 添加授权，按角色/组授权，提交 已选角色/组，待添加的人员，待移除的人员

```
POST /v1/admin/granted/grantedRoleAccounts
```

```
{
  "operateAccount": "1",
  "grantExpiredDate": "2019-07-23 17:48:00",
  "roleIds": ["1","2","3"],
  "rolegroupIds": ["4","5"],
  "addAccountIds": ["1","2"],
  "delAccountIds": ["3"]
}
```

请求传入 已选的角色 (roleIds) /角色组 (rolegroupIds) , 待添加的人员 (addAccountIds) , 待移除的人员 (delAccountIds)

- 添加授权, 按用户规则授权, 获取 已选用户规则 共有的 角色/组

```
GET /v1/admin/granted/grantedUserscopeRoles
```

```
operateAccount=1&userscopeIds=1,2,3
```

请求传入 用户规则 (userscopeIds) , 查询到用户规则 共同拥有的 角色 (roleIds) 或 角色组 (rolegroupIds) , 返回

- 添加授权, 按用户规则授权, 提交 已选用户规则, 待添加的角色/组, 待移除的角色/组

```
POST /v1/admin/granted/grantedUserscopeRoles
```

```
{
  "operateAccount": "1",
  "grantExpiredDate": "2019-07-23 17:48:00",
  "userscopeIds": ["1","2","3"],
  "addRoleIds": ["1","2"],
  "addRolegroupIds": ["3"],
  "delRoleIds": ["9"],
  "delRolegroupIds": ["4","5"]
}
```

请求传入 已选的用户规则 (userscopeIds) , 待添加的角色 (addRoleIds) 或 角色组 (addRolegroupIds) , 待移除的角色 (delRoleIds) 或 角色组 (delRolegroupIds) (一般为共同拥有的)

更新 用户规则 与 角色 或 角色组 的授权关系

分级授权

- 分页查询分级授权的管理帐号

```
GET /v1/admin/manGrantedAccounts
```

```
operateAccount=1&loadAll=false&pageIndex=0&pageSize=20&mapBean[keyword]=帐号/姓名  
&mapBean[identityType]=教师
```

- 根据 id 获取分级授权的详情

```
GET /v1/admin/manGrantedAccounts/{id}
```

```
operateAccount=1
```

响应：

```
{  
  "data": {  
    "id": "1",  
    "accountId": "1",  
    "username": "T000001",  
    "name": "张校办",  
    "identityType": "教师",  
    "organizationName": "校办",  
    "state": "正常",  
    "manGrantedAccountRoles": [  
      {  
        "id": "1",  
        "accountId": "1",  
        "roleType": "Role",  
        "rolePk": "1",  
        "canGrant": true,  
        "canDataGrant": false,  
        .....  
      },  
      {  
        "id": "2",  
        "accountId": "1",  
        "roleType": "Rolegroup",  
        "rolePk": "2",  
        "canGrant": true,  
        "canManGrant": true,  
        .....  
      }  
    ],  
    .....  
  }  
}
```

- 添加分级授权

```
POST /v1/admin/manGrantedAccounts/roles

{
  "operateAccount": "1",
  "grantExpiredDate": "2019-07-23 17:48:00",
  "accounts": [
    {
      "accountId": "1",
      "username": "T000001",
      "name": "张校办",
      "identityType": "教师",
      "organizationName": "校办",
      "state": "正常"
    },
    {
      "accountId": "2",
      "username": "T000002",
      "name": "李后勤",
      "identityType": "教师",
      "organizationName": "后勤",
      "state": "正常"
    }
  ],
  "manGrantedAccountRoles": [
    {
      "roleType": "Role",
      "rolePk": "1",
      "canGrant": true,
      "canManGrant": false
    },
    {
      "roleType": "Rolegroup",
      "rolePk": "2",
      "canGrant": true,
      "canManGrant": true
    }
  ]
}
```

请求传入 已选的 帐号 (accounts) , 待添加的 角色 (manGrantedAccountRoles)

嵌套循环 已选的 帐号, 待添加的 角色, 保存 ManGrantedAccount、ManGrantedAccountRole

- 修改分级授权

```
PUT /v1/admin/manGrantedAccounts/{id}/roles
```



```
{
  "operateAccount": "1",
  "grantExpiredDate": "2019-07-23 17:48:00",
  "manGrantedAccountRoles": [
    {
      "roleType": "Role",
      "rolePk": "1",
      "canGrant": true,
      "canManGrant": false
    },
    {
      "roleType": "Rolegroup",
      "rolePk": "2",
      "canGrant": true,
      "canManGrant": true
    }
  ]
}
```

- 替换分级授权的管理帐号

PUT /v1/admin/manGrantedAccounts/{id}/replace

```
{
  "operateAccount": "1",
  "account": {
    "accountId": "2",
    "username": "T000002",
    "name": "李后勤",
    "identityType": "教师",
    "organizationName": "后勤",
    "state": "正常"
  }
}
```

请求传入，待替换的管理帐号信息，进行替换

替换，需要处理以下事项：

1. 新增 ManGrantedAccount，但须判断 新 accountId 是否已经存在；若存在 新 accountId 的 ManGrantedAccount，则须判断 其的 grantAccount 是否为 原 accountId，若 grantAccount 与 原 accountId，则 **提示无法替换**
2. 将 ManGrantedAccountRole 对应的 accountId 替换为新的管理帐号（accountId），并将原来由 该管理帐号 处理的分级授权数据 全部替换，ManGrantedAccount，ManGrantedAccountRole 表中的 grantAccount，revokeAccount

- 撤销分级授权

```
PUT /v1/admin/manGrantedAccounts/{id}/revoke
```

```
{
  "operateAccount": "1"
}
```

请求传入 分级授权 id 进行撤销

撤销，需要处理以下事项：

1. 将 该管理帐号（ManGrantedAccount 的 accountId）处理的授权数据（ManGrantedAccountRole 的 roleType、rolePk）的记录 全部撤销，GrantedAccountRole, GrantedUserscopeRole, GrantedAccountRolegroup, GrantedUserscopeRolegroup 表中的 grantAccount 为 accountId, roleId或rolegroupId 对应的记录进行撤销。并且 还须 追溯所有分级授权的管理帐号 处理的授权数据，进行撤销
2. 将 该管理帐号（ManGrantedAccount 的 accountId）处理的分级授权数据（ManGrantedAccountRole 的 roleType、rolePk）的记录 全部撤销，ManGrantedAccount, ManGrantedAccountRole 表中的 grantAccount 为 accountId, roleType、rolePk 对应的记录进行撤销。并且 还须 追溯所有分级授权数据，进行撤销
3. 根据 id, 将 ManGrantedAccount 进行撤销，将关联的 ManGrantedAccountRole 进行撤销；

授权日志

- 分页查询授权操作日志
- 分页查询授权访问日志

授权统计

接口定义

详见 [User Authorization Service Super Admin APIs \(v1\)](#)

开放接口 poa

功能说明

用户角色

- 获取用户的角色

接口定义

详见 [User Authorization Service Platform Open APIs \(v1\)](#)