# 设计背景

## 1.3 授权模块

授权模块是确认通过身份验证的用户可以进行哪些操作的模块。此模块中，保存有系统内所有功能的列表。同时，拥有角色实体，可以通过角色管理器，查询到用户属性的角色。

[请补充用例图]

### 1.2.1 添加角色

向角色表中添加一个角色

网络接口：/api/authorize/addRole

参数：

token：登录令牌，可以是用户名，手机号，邮箱或者其它可以识别用户身份的字段。

name: 角色的名称

resources: 角色可以访问的资源列表，以半角的逗号分隔

输出：

success：登录是否成功

msg：登录失败的原因

### 1.2.2 查询角色列表

从数据库中查询所有角色。

网络接口：/api/authorize/getRoles

参数：

\_accessToken\_：登录凭据

page：页码，默认为1

pageSize：每页条数，默认为10

data：数组，其中每一项为一个角色。拥有id，name和resources三个字段，对应角色表中的同名属性

### 1.2.3 删除角色

[请自行补充]

网络接口：/api/authorize/removeRole

参数：

\_accessToken\_：登录凭据

id：待删除的角色ID

### 1.2.4 分配和取消角色

为用户分配角色。

网络接口：/api/authorize/grant

参数：

\_accessToken\_：登录凭据

userId：角色ID

roleIds：被分配的角色ID，用半角逗号隔开

### 1.2.5 获取权限

通过UserId获取该用户可以执行的所有权限列表。先获取用户对应的角色，再获取角色对应的权限。最后将其组合在一起。

网络接口：无

提示：在用户处，必须补充一个获取角色的接口。即可以是单独的网络接口，也可以在“获取用户”这一用例的返回值里，补充角色列表。

# 第四章 授权模块设计

身份验证模块包含授权服务，资源管理器，角色管理器，资源类，角色类，用户角色关联类等组成，类图如下：

图示

描述已自动生成

图4-1 授权模块类图

[注：关系图中的IUserPrincipal和User是其它章节中已经编写好的成品类，属于模块的外部。在编写下方的文档时，请不要将这两个类重复写入]

## 3.1 AuthorizationService

该类负责所有授权服务。其下包含两个私有的管理器：资源管理器和角色管理器。其中，资源管理器保存有系统中所有的资源（即功能点，比如添加用户，删除用户都是资源），角色管理器负责维护系统中所有的角色。

[其余请自行补充]

## 3.2 ResourceManager

在资源管理器的构造函数中，通过硬编码的方式将系统所有功能加入其中。

[请自行补充]

## 3.2 RoleManager

角色管理器负责管理系统角色，角色是一种数据库实体。需要对其进行增加，删除，查询和修改等操作。

同时，该管理器负责维护角色与用户之间的关系。每个角色都可以拥有若干个用户，每个用户可以同时拥有多个角色。所以，这种关系需要一张关系表用以保存。

[请自行补充]