**设计文档**

新系统使用SpringCloud体系搭建微服务平台。

使用SpringBoot框架开发微服务。

使用Eureka组件实现服务的注册与发现。

使用Feign组件实现服务消费者。

使用Zuul组件实现路由转发，和Ribbon结合实现了负载均衡。

使用Spring Cloud Config实现微服务的分布式配置中心。

使用Hystrix组件实现SpringCloud的断路器功能。

使用JWT和Spring Security实现身份鉴别。

使用JPA实现数据持久化操作。

前端使用Ace Admin模板完成页面开发。

页面使用HTML5，并结合thymeleaf组件静态开发。

SpringBoot开发的微服务搭打包成Docker镜像。

使用Kubernetes平台完成容器集群管理。

使用Gitlab平台管理代码。

1. **权限组织管理**

权限组织管理为平台的基础管理部分，定义了组织机构层次、人员目录以及用户权限。

* 1. **组织人员管理**

组织人员管理包括对组织和人员的管理工作，可增删改查组织机构和人员，每个组织有唯一编码，采用类似01、0101、010101、010102的命名规则方便查询下级组织和人员。组织管理有租户概念，根节点的下级即为租户级别。

* + 1. 获取组织的下级组织和人员列表

获取组织的下级组织和人员，以集合形式返回给接口调用者。

* + 1. 添加下级组织

为当前组织添加下级组织。

* + 1. 添加人员

为当前组织添加所属人员

* + 1. 修改组织

修改组织信息，包括状态（正常、停用），组织编号不许修改。

* + 1. 修改人员

修改人员信息，包括状态（正常、停用），

* + 1. 获取人员信息

查询人员的基本信息，包括人员的角色和权限

* + 1. 获取人员列表

获取人员列表，并以树的形式返回

* 1. **权限管理**

权限管理目前只对功能模块的页面权限进行定义，仅供开发设计人员使用，用户管理员可对权限进行授予。

权限管理以树形展示，树的节点为菜单各级页面，节点下面挂载权限项目。

* + 1. 获取节点下属节点或者权限

获取节点的下属节点和权限，并以集合的形式返回

* + 1. 添加下级节点

为节点添加下级节点

* + 1. 添加所属权限

为节点添加所属的权限

* + 1. 修改节点信息

修改节点的相关信息，包括节点的状态（正常、停用），编号不许修改

* + 1. 修改权限信息

修改权限的相关信息，包括节点的状态（正常、停用），编号不许修改

* + 1. 获取权限列表

获取权限类别，并以树的形式返回

* 1. **角色管理**

角色管理定义系统的可用角色，分为基础角色和定制角色，基础角色为系统定义好的，定制角色为各个租户自行创建。

* + 1. 获取角色列表

获取所有的角色，以集合形式返回

* + 1. 修改角色

修改角色信息，包括状态（正常、停用）

* + 1. 角色与权限绑定

为角色赋予权限集合

* + 1. 角色与人员绑定

把角色赋予人员