**数据表**

domain 产品线管理

role 角色管理

user 用户管理

user\_domain\_role\_relation 用户于角色的关联

domain\_role\_resource\_relation 角色和具体权限管理表

**实现接口：**

**针对uc\_domain表与产品线管理相关的接口：**

/domain/add 添加产品线

参数：domain\_name, domain\_desc, domain\_type=0,domain\_salt，

default\_role\_id=null,status

返回：code 0: 代表成功 -1: 代表失败

msg 返回具体的提示信息

实现逻辑：1. 参数合法性校验,status 1:可用 2：禁用

2. 自动生成时间create\_date update\_time字段值

3. 插入到数据库domain表中

4. 返回结果

/domain/list 查询产品线

参数：void

返回：

id为空，返回id不能为空，

如果is\_delete = 1，返回该产品线已下线

没有问题返回以下字段：

domain\_name, domain\_desc, domain\_type,

default\_role\_id,status

实现逻辑：

1. 从数据库检索所有记录，不包含is\_delete 为1的

2. 返回结果

/domain/update 更新产品线

参数：id, domain\_name, domain\_desc, domain\_type,

default\_role\_id,status，is\_delete

Update table set name=$\_post[‘name’] where id=’’

返回：code 0: 代表成功 -1: 代表失败

msg 返回具体的提示信息

实现逻辑：1. 参数合法性校验

2. 更新数据到数据表

3. 返回结果

/domain/delete 删除产品线, 不是真正的删除，只是修改一下数据表 is\_delete的状态

参数：id 1,2,3,4 4

返回：code 0: 代表成功 -1: 代表失败

msg 返回具体的提示信息

实现逻辑：1. 验证id是否为空，验证该产品线是否存在，是否已经删除

2. 更新数据库将is\_delete值改为1

3. 返回结果

**针对uc\_user表与用户管理相关的接口：**

/user/add 添加用户

参数：network\_type, name, image, email, status

返回：code 0: 代表成功 1: 代表失败

msg 返回具体的提示信息

实现逻辑： 1. 参数合法性校验,检查邮箱格式，status只能是1 （可用）,0

（禁用）

2. 插入到数据库中

3. 返回结果

/user/list 查询用户

参数：id

返回：

id为空，返回id不能为空，

如果is\_delete = 1，返回该用户已被删除

没有问题返回以下字段：

network\_type, name, image, email, status, is\_delete

实现逻辑：1. 参数合法性验证，看是id是否为空

2. 从数据库检索信息，验证结果集是否为空，检验is\_delete 是否

为1

3. 返回结果

/user/update 更新用户

参数：id, network\_type, name, image, email,

status

返回：code 0: 代表成功 1: 代表失败

msg 返回具体的提示信息

实现逻辑：1. 参数合法性校验，检查email格式，status只能是1 （可用）,0

（禁用）

2. 更新数据到数据表

3. 返回结果

/user/delete 删除用户, 不是真正的删除，只是修改一下数据表 is\_delete的状态

参数：id

返回：code 0: 代表成功 -1: 代表失败

msg 返回具体的提示信息

实现逻辑：1. 验证id是否为空，验证该用户是否存在，是否已经删除

2. 更新数据库将is\_delete值改为1

3. 返回结果

**针对uc\_role表与角色管理相关的接口：**

/role/add 添加角色

参数：domain\_id, role\_name, role\_desc, role\_type, [status]

返回：code 0: 代表成功 -1: 代表失败

msg 返回具体的提示信息

实现逻辑： 1. 参数合法性校验,验证domain\_id的有效性，role\_type只能是

1（内网）2（外网）

2. 自动生成时间create\_date update\_time字段值

3. 插入到数据库中

4. 返回结果

/role/list 查询角色

参数：id

返回：

id为空，返回id不能为空，

如果is\_delete = 1，返回该角色已被删除

没有问题返回以下字段：

Domain\_id, role\_name, role\_desc, role\_type, status

实现逻辑：1. 参数合法性验证，看是id是否为空

2. 从数据库检索信息，检验is\_delete 是否为1

3. 返回结果

/role/update 更新角色

参数：id,role\_name, role\_desc, role\_type, status

返回：code 0: 代表成功 -1: 代表失败

msg 返回具体的提示信息

Domain\_id 不允许修改

实现逻辑：1. 参数合法性校验，验证domain\_id的有效性，role\_type只能是 1

（内网）2（外网)

2. 更新数据到数据表

3. 返回结果

/role/delete 删除角色, 不是真正的删除，只是修改一下数据表 is\_delete的状态

参数：id

返回：code 0: 代表成功 -1: 代表失败

msg 返回具体的提示信息

实现逻辑：1. 验证id是否为空，验证该角色是否存在，是否已经删除

2. 更新数据库将is\_delete值改为1

3. 返回结果

**针对uc\_resource表与角色权限管理相关的接口：**

/resource/add 添加权限

参数：domain\_id, resource\_url, resource\_name,resource\_desc, [status ]

返回：code 0: 代表成功 -1: 代表失败

msg 返回具体的提示信息

实现逻辑：

1. 参数合法性校验,验证domain\_id的有效性，resource\_url的url

格式,status只能是1（内网）2（外网）

2. 插入到数据库中

3. 返回结果

/resource/list 查询权限

参数： id

返回：

id为空，返回id不能为空，

如果is\_delete = 1，返回该权限已被删除

没有问题返回以下字段：

domain\_id, resource\_url, resource\_name,resource\_desc, status

实现逻辑：1. 参数合法性验证，检验id是否为空

2. 从数据库查询记录，验证该权限是否存在且没有

被删除

3. 返回结果

/resource/update 更新权限

参数：id, resource\_url, resource\_name,resource\_desc, [status ]

返回：code 0: 代表成功 -1: 代表失败

msg 返回具体的提示信息

实现逻辑：

1. 参数合法性校验，验证domain\_id的有效性, 验证resource\_url的

url格式,字段不能为空

2. 更新数据到数据库

3. 返回结果

/resource/delete 删除权限, 不是真正的删除，只是修改一下数据表 is\_delete的状态

参数：id

返回：code 0: 代表成功 -1: 代表失败

msg 返回具体的提示信息

实现逻辑：1. 验证id是否为空，验证该权限是否存在，是否已经删除

2. 更新数据库将is\_delete值改为1

3. 返回结果

/grant/add\_role 给某个用户授权某个角色

参数：user\_id, role\_id

返回：code 0: 代表成功 -1: 代表失败

msg 返回具体的提示信息

实现逻辑：1. 验证user\_id,role\_id是否为空，验证该用户或者角色是否存在，

是否已经删除， 是否可用

2. $domain\_id 等于id = role\_id时查询出uc\_role的domain\_id值，

user\_id,role\_id，domain\_id = $domain\_id 插入数据库的

user\_domain\_role\_relation表

3. 返回结果

/grant/del\_role 给某个用户回收某个角色

参数：user\_id, role\_id

返回：code 0: 代表成功 -1: 代表失败

msg 返回具体的提示信息

实现逻辑：1. 验证user\_id,role\_id是否为空，验证该用户或者角色是否存在，

是否已经删除，确认该用户确实有该角色

2. 删除user\_domain\_role\_relation表中指定user\_id, role\_id的记 录

3. 返回结果

/grant/add\_resource 给某个角色授权某个权限

参数：role\_id,resource\_id

返回：code 0: 代表成功 -1: 代表失败

msg 返回具体的提示信息

实现逻辑：1. 验证role\_id,resource\_id是否为空，验证该角色

或者权限是否存在，是否已经删除

2. $resource\_url 等于id = resource\_id时查询出的resource\_url值， $domain\_id 等于id = role\_id时查询出uc\_role的domain\_id值 domain\_id=$domain\_id, role\_id,resource\_id，resource\_url =

$resource\_url 插入数据库的domain\_role\_resource\_relation表

3. 返回结果

/grant/del\_resource 给某个角色回收某个权限

参数： role\_id,resource\_id

返回：code 0: 代表成功 -1: 代表失败

msg 返回具体的提示信息

实现逻辑：1. 验证role\_id,resource\_id是否为空，验证该角色或者权限是否存

在，是否已经删除，确认该角色确实有该权限

2. 删除domain\_role\_resource\_relation表中指定role\_id,resource\_id

的记录

3. 返回结果

查询接口

/api/resources 查询某个用户在某个产品线下的所有权限

参数： user\_id，domain\_id

返回： 数组：domain\_name和下面的resource\_url

实现逻辑：1. 验证user\_id是否为空，验证该用户是否存在，是否已经删除

2. 验证 domain\_id

2. 从user\_domain\_role\_relation表中利用 user\_id 和 domain\_id 作为条件查询出所有role\_id

在根据role\_id从domain\_role\_resource\_relation表查出

resource\_id的值赋给$resource\_id数组resource\_url

3. 返回结果

/api/roles 查询某个用户在某个产品线下的所有角色

参数： domain\_id， user\_id

返回： role\_name数组

实现逻辑：1. 验证domain\_id是否为空，验证该产品线是否存在，是否已经

删除

2. 根据 domain\_id和user\_id 作为条件查询出所有的 role\_id 在user\_domain\_role\_relation表中，然后联表查询所有的角色信息 ，查询uc\_role表

3. 返回结果

/api/has\_resource 查询某个用户在某个产品线下是否有某个权限

参数： user\_id,domain\_id， url

返回：code 0: 有 -1: 没有

msg 返回具体的提示信息

实现逻辑：1. 实现逻辑和/api/resources基本一致，仅需要判断该url是否在该用户该产品线下的所有权限列表中

3. 返回结果