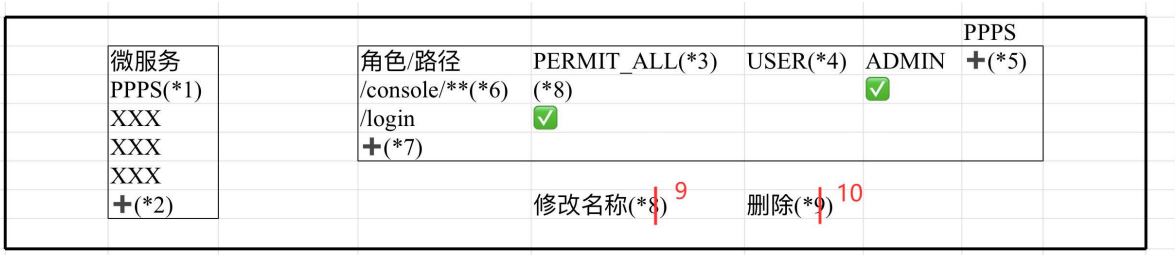
ZAS接口设计

ZAS前端

- 控制台
  - 微服务管理
  - 角色/路径表管理
- 用户页
  - 注册
  - 登录
  - 忘记密码

1 前端界面示例

前端页面实例与页面元素属性说明



*1点击事件	跳转到此微服务	的权限（角色/路径）表
*2点击事件	新建微服务	
*3	允许所有用户访	问的非鉴权路径
	唯一静态数据！	不可点击，不可修改
*4点击事件	修改角色名 或	删除此角色
*5点击事件	新建角色	
*6点击事件	修改路径 或	删除此路径
*7点击事件	新建路径	
*8点击事件	修改微服务名	删除此微服务

2 控制台

2.1 微服务及资源权限管理

Response body描述不够精确，可以用postman测试接口实际功能。

2.1.1 获取全部微服务

用于初始化`*1`所在列

```
url: "/microservice/all",
type: "GET"
```

返回对象列表

2.1.2 获取某个微服务的角色

用于初始化`*3`所在行，`*1`点击事件的一部分

```
url: "/role/by/{msId}"
type: "GET"
```

返回对象列表（或者可以直接拿所有角色，但更建议用上面的）

```
url: "/role/all"
type: "GET"
```

2.1.3 获取某个微服务的资源路径

用于初始化`*6`所在列，`*1`点击事件的一部分

```
url: "/url/by/{msId}"
type: "GET"
```

返回对象列表（或者可以直接拿所有路径，但更建议用上面的）

```
url: "/url/all"
type: "GET"
```

2.1.4 获取权限(角色/路径)表

用于初始化权限(角色/路径)表`*8`，`*1`点击事件的一部分

```
url: "/authority/by/{msId}"
type: "GET"
```

返回对象列表（或者可以直接拿所有表，但更建议用上面的）

```
url: "/authority/all"
type: "GET"
```

2.1.5 新建微服务

用于`*2`的点击事件

```
url: "/microservice"
type: "POST"
data: {
    "name": "developer's microservice name",
}
```

返回（后台自定义）处理结果，如果成功，还会返回新建对象

注：后台自定义处理结果：

- INVALID： 字段不完整或字面值是空串
- EXIST： 字面值冲突
- NOT_EXIST： ID不存在
- PASS：成功通过

2.1.6 新建角色

用于`*5`的点击事件

```
url: "/role"
type: "POST"
data: {
    "msId": "developer's microservice id",
    "name": "role name",
}
```

返回处理结果，如果成功，还会返回新建对象

2.1.7 修改角色名

用于`*4`点击事件（如弹窗）后的按钮点击事件

```
url: "/role/update"
type: "POST"
data: {
    "id": "role id",
    "name": "latest role name",
}
```

返回处理结果

2.1.8 删除角色

用于`*4`点击事件（如弹窗）后的按钮点击事件

```
url: "/role/delete"
type: "POST"
data: {
    "id": "role id",
}
```

返回处理结果

2.1.9 新建路径

用于`*7`的点击事件

```
url: "/url"
type: "POST"
data: {
    "msId": "developer's microservice id",
    "path": "url path that does not contain context path",
}
```

返回处理结果，如果成功，还会返回新建对象

2.1.10 修改路径名

用于`*6`点击事件（如弹窗）后的按钮点击事件

```
url: "/role/update"
type: "POST"
data: {
    "id": "url id",
    "path": "latest url path",
}
```

返回处理结果

2.1.11 删除路径

用于`*6`点击事件（如弹窗）后的按钮点击事件

```
url: "/role/delete"
type: "POST"
data: {
    "id": "url id",
}
```

返回处理结果

2.1.12 增加角色-路径关系

用于`*8`点击事件

```
url: "/authority"
type: "POST"
data: {
    "msId": "developer's microservice id",
    "urlId": "url (row) id",
    "roleId": "role (column) id",
}
```

返回处理结果，如果成功，还会返回新建对象

2.1.13 删除角色-路径关系

用于`*8`点击事件

```
url: "/authority/delete"
type: "POST"
data: {
    "id": "authority id"
}
```

返回处理结果

2.1.14 修改微服务名称

用于`*9`点击事件

```
url: "/microservice/update"
type: "POST"
data: {
    "id": "developer's microservice id",
    "name": "latest microservice name"
}
```

返回处理结果

2.1.15 删除微服务

用于`*10`点击事件

```
url: "/microservice/delete"
type: "POST"
data: {
    "id": "developer's microservice id",
}
```

返回处理结果

2.2 用户注册管理

为某种途径注册的用户分配某些角色（批量管理未来注册的用户），为此需要设计注册管理视图

2.2.1 获取所有注册

用于初始化注册管理视图

```
url: "/signup/all"
type: "GET"
```

返回全部注册对象

3.2.2 新建注册

```
url: "/signup"
type: "POST"
data: {
    "name": "signup's friendly name",
}
```

返回处理结果，如果成功，还会返回新建对象

2.2.3 修改注册的友好名称

```
url: "/signup/update"
type: "POST"
data: {
    "id": "signup id",
    "name": "signup's friendly name",
}
```

返回处理结果

2.2.4 删除注册

```
url: "/signup/delete"
type: "POST"
data: {
    "id": "signup id",
}
```

返回处理结果

2.2.5 为注册分配角色

从注册或角色上添加点击事件

```
url: "/signup/add_role"
type: "POST"
data: {
    "id": "signup id",
    "roleId": "role id",
}
```

返回处理结果

2.2.6 为注册移除角色

从注册或角色上添加点击事件

```
url: "/signup/remove_role"
type: "POST"
data: {
    "id": "signup id",
    "roleId": "role id",
}
```

返回处理结果

2.3 用户角色管理

为用户分配角色（独立管理已注册的用户），为此需要设计用户管理视图

2.3.1 获取全部用户-角色

```
url: "/user_role/all"
type: "GET"
```

返回全部用户-角色对象

2.3.2 为用户分配角色

```
url: "/user_role"
type: "POST"
data: {
    "userId": "username",
    "roleId": "role id",
}
```

返回处理结果，如果成功，还会返回新建对象

2.3.3 为用户移除角色

```
url: "/user_role/delete"
type: "POST"
data: {
    "id": "user-role id"
}
```

返回处理结果

3 用户页

3.1 注册

用户注册接口，为此需要设计用户注册视图，多个注册视图是可行的。

```
url: "/auth/signup"
type: "POST"
headers: {
    "username": "user's id that is unique and immutable",
    "password": "user's password",
    "signupId": "signup id",
}
```

返回处理结果

3.2 登录

用户登录接口，为此需要设计用户登录视图，多个登录视图时可行的。

```
url: "/auth/signin"
type: "POST"
headers: {
    "username": "user's id",
    "password": "user's password",
}
```

如果成功返回口令

3.3 修改密码

现系统不登记邮箱或手机，缺少必要安全认证途径，用户如修改密码需联系管理员。

附注 对象

```
public class Authority {
    private String id;
    private String msId;
    private String urlId;
    private String roleId;
}
```

```
public class Microservice {
    private String id;
    private String name;
}

public class Role {
    private String id;
    private String msId;
    private String name;
}

public class URL {
    private String id;
    private String msId;
    private String path;
}
```