|  |
| --- |
| 此框内是补充信息，可能会对前面的内容有修改 |

**（1）显示当前页面用户信息**

1）post

2）所需权限：所有用户

3）向服务器传递4个参数：

pager.pageNo=2&pager.pageSize=5&sort=userId&direction=desc

注：pager.pageNo：传递当前是第几页； pager.pageSize：传递当前每页多少条记录； sort：传递当前排序的字段名； direction：传递当前的排序方向,dsec表示降序

4）服务器返回JSON数据：

{"pager.pageNo":2,"pager.totalRows":23,"rows":[

{},

{}，

···

]}

注1：rows元素个数与pager.pageSize一致；pager.totalRows：总数据条数（而非当前返回的数据条数）

注2：

{“**userLoginName**”:“”,“**userPassword**”: “”, “**userName**”: “”, “**userSex**”: “”, “**userOrganization**”: “”, “**userTel**”: “”, “**userEmail**”: “”, “**userLevel**”: “”, “**userEmployTime**”: “”, “**userEducation**”: “”, “**userExtraInfo**”: “”}

|  |
| --- |
| Route: /api/user/getUserInfo2;  注：sort 字段目前只支持id,其他暂不支持。  {“**userLoginName**”:“”,“**userPassword**”: “”, “**userName**”: “”, “**userOrganization**”: “”, “**userTel**”: “”, “**userEmail**”: “”, “**userLevel**”: “”, “**userExtraInfo**”} |

**（2）根据关键词搜索显示当前页面用户信息**

1）post

2）所需权限：所有用户

3）向服务器传递4+2个参数：

pager.pageNo=&pager.pageSize=&sort=&direction= &searchIf= &searchInput=

注: pager.pageSize：传递当前每页多少条记录； sort：传递当前排序的字段名； direction：传递当前的排序方向;

searchIf =

1：查询类型为姓名；

2：查询类型为单位；

3：查询类型为权限等级。

searchInput：搜索的关键词

|  |
| --- |
| Route: api/user/getUserInfo2  权限等级用1,2,3,4表示  权限等级查询方式后端不支持，删除之 |

4）服务器返回JSON数据：

{"pager.pageNo":1,"pager.totalRows":23,"rows":[

{},

{}，

···

]}

注1：rows元素个数与pager.pageSize一致；pager.totalRows：总数据条数（而非当前返回的数据条数）

注2：

{“**userLoginName**”:“”,“**userPassword**”: “”, “**userName**”: “”, “**userSex**”: “”, “**userOrganization**”: “”, “**userTel**”: “”, “**userEmail**”: “”, “**userLevel**”: “”, “**userEmployTime**”: “”, “**userEducation**”: “”, “**userExtraInfo**”: “”}

**（3）添加用户**

分为三步:

S1:验证用户名的唯一性

S2:获取当前登录用户能够分配的项目

S3:提交表单，返回结果；

S1:

1） GET

2）向服务器传递参数：

validateValue= &\_=1562831119379

（后面那个不知道为啥会有···，只需要validateValue传递新增用户的用户名）

3）返回

{"validateResult":{"valid":true }}

验证通过为true，失败为false

|  |
| --- |
| 还要验证邮箱和手机号是否重复（算了，不验证了）但是，应该返回一些错误信息，比如使用Toast. 这要做。 |

S2:

1. POST
2. 向服务器传递参数：sysUserId
3. 返回

{"treeNodes":[{ "id":"1", "parentId":"0", "name":"项目1", "open": "true"},{ "id":"2", "parentId":"0", "name":"项目2"},{ "id":"3", "parentId":"0", "name":"项目3"}]}

备注："parentId"必须全为0

S3:

1）post

2）特定用户

3）向服务器传递参数：

sysUserId=&userLoginName=&userPassword=&username=&userSex=&userOrganization=&userTel=&userEmail=&userLevel=&userProject=&userEmployTime=&userEducation=&userExtraInfo=

注：{"value":"1","key":"威腾管理人员"},{"value":"2","key":"区域业务员"},{"value":"3","key":"客户公司管理人员"},{"value":"4","key":"客户公司运维人员"}

4）服务器返回

responseText.**status**

成功返回1；失败返回0；

**（4）修改用户信息**

分为三步:

S1:验证当前登录用户的修改权限

S2:给服务器传递待修改的用户ID，得到修改之前的信息；

S3:提交表单和当前登录用户ID，给服务器传递修改后的信息。

S1：

1）post

2）向服务器传递参数：

userId=&sysUserId=

3）服务器返回

1：有权限

0：无权限

S2：

1）post

2）向服务器传递参数：

userId=

3）服务器返回

{“**userLoginName**”:“”,“**userPassword**”: “”, “**userName**”: “”, “**userSex**”: “”, “**userOrganization**”: “”, “**userTel**”: “”, “**userEmail**”: “”, “**userLevel**”: “”, “**userProject**”: “”“**userEmployTime**”: “”, “**userEducation**”: “”, “**userExtraInfo**”: “”}

S3：

1）post

2）向服务器传递参数：

sysUserId=&userId=&userLoginName=&userPassword=&username=&userSex=&userOrganization=&userTel=&userEmail=&userLevel=&userEmployTime=&userEducation=&userExtraInfo=

3）服务器返回

responseText.**status**

成功返回1；失败返回0；

|  |
| --- |
| S1 : 前端在user\_list.js onEditUser();  后端在 /api/user/validateAuthority  POST, 参数：”userId”,(删除了sysUserId，后端可判断sysUserId，无需) |
|  |

**（5）删除用户**

1）post

2）向服务器传递参数：

userId=&sysUserId=

3）服务器返回

data

成功返回1；失败返回0

|  |
| --- |
| sysUserId无需上传 |

**（6）查看用户详细信息**

分为两步:

S1:验证当前登录用户查看详情的权限

S2:给服务器传递待查看的用户ID，得到用户的信息；

S1：

1）post

2）向服务器传递参数：

userId=&sysUserId=

3）服务器返回

result

1：有权限

0：无权限

S2：

1）post

2）向服务器传递参数：

userId=

3）服务器返回

{“**userLoginName**”:“”,“**userPassword**”: “”, “**userName**”: “”, “**userSex**”: “”, “**userOrganization**”: “”, “**userTel**”: “”, “**userEmail**”: “”, “**userLevel**”: “”, “**userProject**”: “”， “**userEmployTime**”: “”, “**userEducation**”: “”, “**userExtraInfo**”: “”}