# 用户操作接口

## 检测用户名（邮箱）是否存在

### 接口参数

|  |  |
| --- | --- |
| 检测用户名（邮箱）是否存在 | |
| uri | /user/checkEmail |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| email | 用户输入邮箱 | String(20) | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400操作异常 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status":400,

"msg":"邮箱不合法",

"data":null

}

### 开发要求

#### 前端

前端需要对用户输入邮箱进行校验，邮箱格式合法且不为空。

#### 后端

后端需要对用户邮箱进行校验，邮箱格式合法且不为空。

## 注册

### 接口参数

|  |  |
| --- | --- |
| 注册 | |
| uri | /user/register |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| userName | 用户名（邮箱） | String(20) | Y |
| password | 密码 | String(20) | Y |
| nickName | 昵称 | String(20) | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400操作异常 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status":400,

"msg":"邮箱已被注册",

"data":null

}

### 开发要求

#### 前端

前端需要对用户输入邮箱进行校验，邮箱格式合法且不为空，密码不为空且首尾不能包含空格。

#### 后端

后端需要对用户邮箱进行校验，邮箱格式合法且不为空，密码不为空且首尾不能包含空格。

## 登录

### 接口参数

|  |  |
| --- | --- |
| 登录 | |
| uri | /user/login |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| userName | 用户名（邮箱） | String(20) | Y |
| password | 密码 | String(20) | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400操作异常 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status":400,

"msg":"用户名或密码错误",

"data":null

}

### 开发要求

#### 前端

前端需要对用户输入邮箱进行校验，邮箱格式合法且不为空，密码不为空且首尾不能包含空格。

#### 后端

后端需要对用户邮箱进行校验，邮箱格式合法且不为空，密码不为空且首尾不能包含空格，先通过用户名查出用户信息，再比较密码。

## 用户查询

### 接口参数

|  |  |
| --- | --- |
| 用户查询 | |
| uri | /user/findUser |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| nickName | 用户昵称 | String(20) | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400操作异常 | Integer |
| msg | 响应信息 | String |
| data | 相似昵称名的用户信息对象 | Object |

### 返回示例

{

"status": 200,

"msg": "操作成功",

"data": [

{

"userNo": 2,

"nickName": "编号18701",

"introducti n": "个人介绍"

},

{

"userNo": 10,

"nickName": "编号10539",

"introducti n": "个人介绍"

},

{

"userNo": 11,

"nickName": "123",

"introducti n": "个人介绍"

}

]

}

### 开发要求

#### 前端

传入不为空的用户昵称。

#### 后端

从session中判断用户已经登录了，才能进行查询操作。

## 查询个人信息

### 接口参数

|  |  |
| --- | --- |
| 用户查询 | |
| uri | /user/findUserInfo |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

（无）

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400操作异常 | Integer |
| msg | 响应信息 | String |
| data | 用户个人信息对象 | Object |

### 返回示例

  {

"status": 200,

"msg": "操作成功",

"data":

{

"userNo": 2,

"nickName": "编号18701",

"introducti n": "个人介绍"

}

### 开发要求

#### 前端

无。

#### 后端

从session中判断用户已经登录了，才能进行查询操作。

## 用户查询

### 接口参数

|  |  |
| --- | --- |
| 用户查询 | |
| uri | /user/updateNickName |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| newNickName | 新的用户昵称 | String(20) | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400操作异常 | Integer |
| msg | 响应信息 | String |
| data | 修改后的用户信息对象 | Object |

### 返回示例

{

"status": 200,

"msg": "操作成功",

"data":

{

"userNo": 2,

"nickName": "编号18701",

"introducti n": "个人介绍"

}

### 开发要求

#### 前端

传入不为空的用户昵称。

#### 后端

从session中判断用户已经登录了，才能进行修改操作。

## 修改密码

### 接口参数

|  |  |
| --- | --- |
| 修改密码 | |
| uri | /user/updatePassword |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| oldPassword | 旧密码 | String(20) | Y |
| newPassword | 新密码 | String(20) | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400操作异常 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status": 200,

"msg": "操作成功",

"data": null

### 开发要求

#### 前端

传入的旧密码和新密码不能为空。

#### 后端

从session中判断用户已经登录了，才能进行修改操作，修改前校验用户旧密码是否正确。

# 看板空间操作接口

## 查询看板空间详细内容

### 接口参数

|  |  |
| --- | --- |
| 查询看板详细内容 | |
| uri | /boardSpace/selectBoardSpaceByUserId |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

（无）

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400操作失败 | Integer |
| msg | 响应信息 | String |
| data | 看板空间详细信息对象 | Object |

### 返回示例

{

"status": 200,

"msg": "操作成功",

"data": [

{

"boardSpaceNo": 41, // 看板空间编号

"boardSpaceName": "个人面板", // 看板空间名

"summary": "个人面板", // 看板空间摘要

"categoryNo": 1, // 看板空间类型1：个人看板2：普通项目

"userNo": 11, // 看板空间所有人

"tBoard": [ // 看板空间中的看板，顺序已经排好

{

"boardNo": 4, // 看板编号

"boardName": "2$3编号看板", // 看板名称

"displayNo": 1, // 看板可见性1：仅看板成员可见2：仅项目成员可见

"startTime": 1504774237000, // 看板起始时间

"endTime": 1506070237000, // 看板截至时间

"boardSpaceNo": 2, // 看板所属看板空间编号

"isDelete": false, // 是否已经被删除

"userNo": 3, // 看板所有人

"listOrder": null // 看板中列表的顺序

},

{

"boardNo": 1,

"boardName": "1$1编号看板",

"displayNo": 1,

"startTime": 1504774236000,

"endTime": 1506070236000,

"boardSpaceNo": 1,

"isDelete": false,

"userNo": 1,

"listOrder": "1,2"

}

]

}

]

}

### 开发要求

#### 前端

（无）

#### 后端

session中存在用户编号，过滤掉isdelete已删除数据，返回的看板数据有顺序。

## 2、新增看板空间

### 接口参数

|  |  |
| --- | --- |
| 新增看板 | |
| uri | /boardSpace/addBoardSpace |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| boardSpaceName | 新建看板空间名称 | String(70) | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法，500服务器异常 | Integer |
| msg | 响应信息 | String |
| data | 新建的看板空间信息 | Object |

### 返回示例

{

"status": 200,

"msg": "OK",

"data": {

"boardSpaceNo": 42, // 看板空间编号

"boardSpaceName": "些钱", // 看板空间名

"summary": "些钱", // 看板空间摘要

"categoryNo": 2, // 看板空间类型1：个人空间2：普通项目

"userNo": 11, // 看板空间所有人

"isDelete": false, // 看板空间是否已经被删除

"boardOrder": null // 看板空间中看板顺序

}

}

### 开发要求

#### 前端

新建看板空间名称不能为空。

#### 后端

session中存在用户编号，且用户存在，新建看板空间名不能为空，对新建看板空间名进行html实体转换，看板空间类型为普通项目。

## 3、删除看板空间

### 接口参数

|  |  |
| --- | --- |
| 删除看板 | |
| uri | /boardSpace/deleteBoardSpace |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| boardSpaceNo | 待删除看板空间编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法，500服务器异常 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status":400,

"msg":"无法删除个人面板",

"data": null

}

### 开发要求

#### 前端

请求参数看板空间编号不能为空。

#### 后端

session中存在用户编写，且用户是看板空间所有人，设置isdelete为true表示删除即可。

## 4、修改看板空间信息

### 接口参数

|  |  |
| --- | --- |
| 修改看板顺序 | |
| uri | /boardSpace/updateBoardSpace |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| boardSpaceNo | 看板空间编号 | Integer | Y |
| newBoardSpaceName | 新看板空间名 | String | Y |
| newSummary | 新看板空间摘要 | String | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法，500服务器异常 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status":400,

"msg":"无法修改个人面板",

"data": null

}

### 开发要求

#### 前端

新看板空间名称不能为空或全为空格，看板空间编号不能为空。

#### 后端

session中存在用户编号，且用户是看板空间所有人，看板isdelete不能等于true，新的看板空间名不可以为空，看板空间属于普通项目，新看板空间名和新摘要都需要进行html实体转换。

## 5、添加看板空间成员

### 接口参数

|  |  |
| --- | --- |
| 添加看板空间成员 | |
| uri | /boardSpace/addUser |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| boardSpaceNo | 看板空间编号 | Integer | Y |
| targetUserNo | 待添加用户编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status":200,

"msg":"操作成功",

"data": null

}

### 开发要求

#### 前端

看板空间编号和待添加用户编号不为空。

#### 后端

session中存在用户编号，用户是看板空间所有人或看板空间成员。

## 6、删除看板空间成员

### 接口参数

|  |  |
| --- | --- |
| 删除看板空间成员 | |
| uri | /boardSpace/deleteUser |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| boardSpaceNo | 看板空间编号 | Integer | Y |
| targetUserNo | 待删除用户编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status":200,

"msg":"操作成功",

"data": null

}

### 开发要求

#### 前端

看板空间编号和待删除用户编号不为空。

#### 后端

session中存在用户编号，用户是看板空间所有人，看板空间isdelete不能为true，待移除用户不是看板所有人。

## 7、查询看板空间成员

### 接口参数

|  |  |
| --- | --- |
| 查询看板空间成员 | |
| uri | /boardSpace/findUser |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| boardSpaceNo | 看板空间编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法 | Integer |
| msg | 响应信息 | String |
| data | 看板空间成员信息 | Object |

### 返回示例

{

"status":200,

"msg":"操作成功",

"data": [

{

"userNo": 2,

"nickName": "编号18701",

"introducti n": "个人介绍"

},

{

"userNo": 10,

"nickName": "编号10539",

"introducti n": "个人介绍"

},

{

"userNo": 11,

"nickName": "123",

"introducti n": "个人介绍"

}

]

}

### 开发要求

#### 前端

看板空间编号不为空。

#### 后端

session中存在用户编号，用户是看板空间所有人或看板空间成员，看板空间isdelete不能为true。

# 看板操作接口

## 1、查询看板详细内容

### 接口参数

|  |  |
| --- | --- |
| 查询看板详细内容 | |
| uri | /board/findBoard/{boardNo} |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| boardNo | 看板编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法，403无权限操作 | Integer |
| msg | 响应信息 | String |
| data | 看板详细信息对象 | Object |

### 返回示例

{

"status": 200,

"msg": "查询成功",

"data": {

"boardNo": 1, // 看板编号

"boardName": "1$1编号看板", // 看板名称

"displayNo": 1, // 可见权限类型

"startTime": 1504774236000, // 看板起始时间

"endTime": 1506070236000, // 看板截至时间

"boardSpaceNo": 1, // 看板所属看板空间编号

"userNo": 1, // 看板创建者

"lists": [ // 看板中列表集合,顺序已经排好

{

"listNo": 1, // 列表编号

"listName": "1", // 列表名称

"cards": [ // 列表中的卡片,顺序已经排好

{

"cardNo": 1, // 卡片编号

"cardTitle": "1", // 卡片标题

"workLoad": 1, // 卡片工作量,为null则不显示

"listNo": 1, // 所属列表编号

"endTime": 1505407959000, // 卡片截至时间,为null则不显示

"tags": [ // 卡片上的标签

{

"tagNo": 1, // 标签编号

"tagName": "1", // 标签名称

"cardNo": 1, // 标签所属卡片编号

"boardNo": 1 // 标签所属看板编号

},

{

"tagNo": 2,

"tagName": "2",

"cardNo": 1,

"boardNo": 2

}

]

}

]

},

{

"listNo": 2,

"listName": "2",

"cards": [

{

"cardNo": 2,

"cardTitle": "2",

"workLoad": 2,

"listNo": 2,

"endTime": 1505407967000,

"tags": []

}

]

}

]

}

}

### 开发要求

#### 前端

请求参数是拼接到url上以post方式请求接口，请求参数看板编号不能为空，且必须为合法数字。

#### 后端

通过session校验用户是否登录，获取传入参数看板编号，确保参数合法性。校验用户是否有权限对看板进行查询操作。

## 2、新增看板

### 接口参数

|  |  |
| --- | --- |
| 新增看板 | |
| uri | /board/addBoard |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| newBoardName | 新建看板名称 | String(70) | Y |
| boardSpaceNo | 看板空间编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法，500服务器异常 | Integer |
| msg | 响应信息 | String |
| data | 新建的看板信息 | Object |

### 返回示例

{

"status":400,

"msg":"用户名或密码错误",

"data": {

boardNo: 1,

boardName: "看板名称",

displayNo: 1,

startTime: 1504774236000,

endTime: 1504774236000,

boardSpaceNo: 1,

isDelete: false,

userNo: 1,

listOrder: null,

}

}

### 开发要求

#### 前端

新建看板名称不能为空。

#### 后端

session中存在用户编号，用户是看板空间成员或看板空间所有人，新建看板名不能为空，对新建看板名进行html实体转换，新建成功后修改看板空间表中的看板顺序。新建成功返回看板信息。

## 3、删除看板

### 接口参数

|  |  |
| --- | --- |
| 删除看板 | |
| uri | /board/deleteBoard |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| boardNo | 看板编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法，500服务器异常 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status":400,

"msg":"用户名或密码错误",

"data": null

}

### 开发要求

#### 前端

请求参数看板编号不能为空。

#### 后端

session中存在用户编号，且用户是看板空间所有人或看板所有人，设置isdelete为true表示删除即可。

## 4、修改看板顺序

### 接口参数

|  |  |
| --- | --- |
| 修改看板顺序 | |
| uri | /board/updateBoardOrder |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| boardOrder | 原看板顺序 | String | Y |
| newBoardOrder | 新看板顺序 | String | Y |
| boardSpaceNo | 看板所属看板空间编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法，500服务器异常 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status":200,

"msg":"修改面板顺序成功",

"data": null

}

### 开发要求

#### 前端

提供有效的原看板顺序以及新看板顺序（格式：[看板编号1],[看板编号2],…，顺序格式为看板编号按顺序排列，用英文逗号分隔），提供有效的看板空间编号。

有效即为不为空。

#### 后端

session中存在用户编号，看板的isdelete不能等于true，且用户是看板空间成员，原看板顺序与数据库中相符，新看板顺序和原看板顺序仅移动看板不同。

## 5、添加看板成员

### 接口参数

|  |  |
| --- | --- |
| 添加看板成员 | |
| uri | /board/addUser |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| boardNo | 看板编号 | Integer | Y |
| targetUserNo | 待添加用户编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status":200,

"msg":"操作成功",

"data": null

}

### 开发要求

#### 前端

看板编号和待添加用户编号不为空。

#### 后端

session中存在用户编号，操作用户是看板空间所有人或看板成员或看板所有人、看板和看板空间isdelete不为true，待添加用户存在且不为看板成员。

## 6、删除卡片成员

### 接口参数

|  |  |
| --- | --- |
| 删除看板成员 | |
| uri | /board/deleteUser |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| boardNo | 看板编号 | Integer | Y |
| targetUserNo | 待删除用户编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status":200,

"msg":"操作成功",

"data": null

}

### 开发要求

#### 前端

看板编号和待删除用户编号不为空。

#### 后端

session中存在用户编号，用户是看板空间所有人或看板成员或看板所有人，看板空间isdelete不能为true，待移除用户不是看板所有人。

如果当前看板是个人空间类型的看板空间，在用户移除成功之后，删除被移除用户个人看板空间表中看板顺序字段中当前看板编号。

移除成功之后去看板空间中查找用户在此看板空间中是否还有关联看板（用户是看板所有人或用户是看板成员），如果没有，且用户不是看板空间所有人，则同步移除看板空间中此用户。

## 7、查询看板成员

### 接口参数

|  |  |
| --- | --- |
| 查询看板成员 | |
| uri | /board/findUser |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| boardNo | 看板编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法 | Integer |
| msg | 响应信息 | String |
| data | 看板成员信息 | Object |

### 返回示例

{

"status":200,

"msg":"操作成功",

"data": [

{

"userNo": 2,

"nickName": "编号18701",

"introducti n": "个人介绍"

},

{

"userNo": 10,

"nickName": "编号10539",

"introducti n": "个人介绍"

},

{

"userNo": 11,

"nickName": "123",

"introducti n": "个人介绍"

}

]

}

### 开发要求

#### 前端

看板编号不为空。

#### 后端

session中存在用户编号，用户是看板空间所有人或看板所有人或看板成员，看板空间和看板的isdelete不能为true。

# 列表操作接口

## 1、修改指定列表名称

### 接口参数

|  |  |
| --- | --- |
| 修改指定列表名称 | |
| uri | /list/updateListName |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| newListName | 更改后列表名称 | String(100) | Y |
| listNo | 列表编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法，403无权限操作 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status": 200,

"msg": "修改列表名称成功",

"data": null

}

### 开发要求

#### 前端

新列表名不能为空，列表编号必须合法。

#### 后端

session中存在用户编号，列表isdelete不能为true，且用户是看板所有人或看板成员，新的列表名不能为空，新列表名需要进行html实体转换。

## 2、新增列表

### 接口参数

|  |  |
| --- | --- |
| 新增列表 | |
| uri | /list/addList |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| listName | 新建列表名称 | String(100) | Y |
| boardNo | 列表所属的看板编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法，500服务器异常 | Integer |
| msg | 响应信息 | String |
| data | 新建的列表信息 | Object |

### 返回示例

{

"status":200,

"msg":"OK",

"data": {

listNo: 1,

listName: "看板名称",

boardSpaceNo: 1,

boardNo: 1,

isDelete: false,

cardOrder: null

}

}

### 开发要求

#### 前端

新建列表名称不能为空，传入的看板编号必须为有效数字。

#### 后端

session中存在用户编号，用户是看板所有人或看板成员，新建列表名不能为空，对新建看板名进行html实体转换，新建成功之后需要修改看板表中的列表顺序。

## 3、删除列表

### 接口参数

|  |  |
| --- | --- |
| 删除列表 | |
| uri | /list/deleteList |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| listNo | 列表编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法，500服务器异常 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status":400,

"msg":"数据不合法",

"data": null

}

### 开发要求

#### 前端

请求参数列表编号不能为空。

#### 后端

session中存在用户编号，且用户是看板所有人或看板成员或者用户是看板空间所有人。

## 4、修改列表顺序

### 接口参数

|  |  |
| --- | --- |
| 修改列表顺序 | |
| uri | /list/updateListOrder |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| listOrder | 原列表顺序 | String | Y |
| newListOrder | 新列表顺序 | String | Y |
| boardNo | 列表所属看板编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法，500服务器异常 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status":200,

"msg":"修改列表顺序成功",

"data": null

}

### 开发要求

#### 前端

提供有效的原列表顺序以及新列表顺序（格式：[列表编号1],[列表编号2],…，顺序格式为列表编号按顺序排列，用英文逗号分隔），提供有效的看板编号。

有效即为不为空。

#### 后端

session中存在用户编号，移动列表isdelete不能为true，且用户属于看板成员，原列表顺序与看板表中数据相符，新列表顺序和原列表顺序仅移动列表不同。

# 卡片操作接口

## 1、修改指定卡片信息

### 接口参数

|  |  |
| --- | --- |
| 修改指定卡片信息 | |
| uri | /card/updateCardInfo |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| newCardName | 更改后卡片名称 | String(4000) | N |
| cardNo | 卡片编号 | Integer | Y |
| workLoad | 工作量 | Double | N |
| endTime | 截至时间（截至时间的毫秒） | Long | N |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法，403无权限操作 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status": 200,

"msg": "修改卡片信息成功",

"data": null

}

### 开发要求

#### 前端

新卡片名、工作量和截至时间，至少传一个有效值。

#### 后端

session中存在用户名，且用户是看板成员，卡片isdelete不为true，新卡片标题、工作量和截至时间都可以为空，新卡片标题需要进行html实体转换。

## 2、新增卡片

### 接口参数

|  |  |
| --- | --- |
| 新增卡片 | |
| uri | /card/addCard |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| cardName | 新建卡片名称 | String(4000) | Y |
| listNo | 列表所属的列表编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法，500服务器异常 | Integer |
| msg | 响应信息 | String |
| data | 新建的卡片信息 | Object |

### 返回示例

{

"status":200,

"msg":"OK",

"data": {

cardNo: 10, // 卡片编号

cardTitle: "xxx", // 卡片标题

workLoad: 0, // 工作量，可以为空，为空不显示

endTime: 1504774236000, // 截止时间，可以为空，为空不显示

boardSpaceNo: 13, // 看板空间编号

boardNo: 12, // 看板编号

listNo: 10, // 列表编号

listName: "列表名", // 列表名，为了方便查询的冗余字段

isDelete: false, // 是否删除标记

}

}

### 开发要求

#### 前端

新建卡片名称不能为空，传入的列表编号必须为有效数字。

#### 后端

session中存在用户编号，用户是看板所有人或看板成员，新建卡片名不能为空，对新建看板名进行html实体转换，创建成功后把修改卡片顺序。

## 3、删除卡片

### 接口参数

|  |  |
| --- | --- |
| 删除卡片 | |
| uri | /card/deleteCard |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| cardNo | 待删除卡片编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法，500服务器异常 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status":400,

"msg":"数据不合法",

"data": null

}

### 开发要求

#### 前端

请求参数卡片编号不能为空。

#### 后端

session中存在用户编号，且用户是看板所有人或看板成员，设置isdelete为true表示删除即可，删除之后需要修改列表表中的卡片顺序。

## 4、修改卡片顺序

### 接口参数

|  |  |
| --- | --- |
| 修改列表顺序 | |
| uri | /card/updateCardOrder |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| cardNo | 移动的卡片编号 | Integer | Y |
| listNoFrom | 卡片所属原列表编号 | Integer | Y |
| listNoTo | 卡片所属新列表编号 | Integer | Y |
| cardOrderFrom | 移动前所属列表中原卡片顺序 | String | Y |
| cardOrderTo | 移动后所属列表中原卡片顺序 | String | Y |
| newcardOrderFrom | 移动前所属列表中新卡片顺序 | String | Y |
| newcardOrderTo | 移动后所属列表中新卡片顺序 | String | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法，500服务器异常 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status":200,

"msg":"修改卡片顺序成功",

"data": null

}

### 开发要求

#### 前端

前端提供有效数据。

#### 后端

session中存在用户编号，卡片isdelete不能为true，且用户均是移动前后的看板成员或看板所有人或看板空间所有人，原卡片顺序与列表表中数据相符，新卡片顺序和原卡片顺序仅移动列表不同。

## 5、查询卡片详细内容

### 接口参数

|  |  |
| --- | --- |
| 查询看板详细内容 | |
| uri | /card/selectBoardSpaceByUserId |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| cardNo | 卡片编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400操作失败，403权限验证失败 | Integer |
| msg | 响应信息 | String |
| data | 卡片详细信息对象 | Object |

### 返回示例

{

"status": 200,

"msg": "操作成功",

"data": {

"cardNo": 1, // 卡片编号

"cardTitle": "1", // 卡片内容

"workLoad": 1, // 工作量，为null表示不现实

"listNo": 11, // 卡片所属列表

"endTime": 1505407959000, // 截至时间，为null表示不显示

"tags": [], // 标签

"comments": [

{

"commentNo": 1,

"comment": "212121212121大胆的", // 评论内容

"userNo": 1, // 评论用户编号

"nickName": "编号45380", // 评论昵称

"commentTime": 1505833778000 // 评论时间

}

]

}

}

### 开发要求

#### 前端

前端提供有效数据。

#### 后端

session中存在用户编号，用户是卡片所在看板的成员或所有人，过滤掉isdelete已删除数据，查询卡片标签，标记用户，评论信息。

## 6、添加卡片成员

### 接口参数

|  |  |
| --- | --- |
| 添加卡片成员 | |
| uri | /card/addUser |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| cardNo | 卡片编号 | Integer | Y |
| targetUserNo | 待添加用户编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status":200,

"msg":"操作成功",

"data": null

}

### 开发要求

#### 前端

卡片编号和待添加用户编号不为空。

#### 后端

session中存在用户编号，用户是看板空间所有人或看板空间成员。

## 删除卡片成员

### 接口参数

|  |  |
| --- | --- |
| 删除卡片成员 | |
| uri | /card/deleteUser |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| cardNo | 卡片编号 | Integer | Y |
| targetUserNo | 待删除用户编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status":200,

"msg":"操作成功",

"data": null

}

### 开发要求

#### 前端

卡片编号和待删除用户编号不为空。

#### 后端

session中存在用户编号，用户是看板空间所有人或看板成员或看板所有人、看板和看板空间isdelete不为true，待添加用户是看板中的成员。

# 评论操作接口

## 1、添加评论

### 接口参数

|  |  |
| --- | --- |
| 添加评论 | |
| uri | /comment/addComment |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| cardNo | 卡片编号 | Integer | Y |
| comment | 评论内容 | String | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法，403无权限操作 | Integer |
| msg | 响应信息 | String |
| data | 评论详细信息对象 | Object |

### 返回示例

  {

"status": 200,

"msg": "操作成功",

"data": {

"commentNo": 10, // 评论编号

"comment": "15325", // 评论内容

"cardNo": 1, // 评论所在卡片编号

"userNo": 11, // 评论用户编号

"commentTime": 1505892506381 // 评论时间

}

}

### 开发要求

#### 前端

评论内容不能为空或全为空格，传入有效的卡片编号。

#### 后端

session中存在用户编号，卡片isdelete不能为true，用户是看板所有人或看板成员，评论内容不能为空，对评论内容进行html实体转换。

## 2、删除评论

### 接口参数

|  |  |
| --- | --- |
| 删除评论 | |
| uri | /comment/deleteComment |
| 请求方式 | post |
| 返回类型 | json |

### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 说明 | 数据类型 | 是否必须 |
| commentNo | 待删除评论编号 | Integer | Y |

### 输出参数

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数据类型 |
| status | 响应状态码，200操作成功，400数据不合法，403权限不足 | Integer |
| msg | 响应信息 | String |
| data | null | null |

### 返回示例

{

"status": 200,

"msg": "操作成功",

"data": null

}

### 开发要求

#### 前端

传入有效的评论编号。

#### 后端

session中存在用户编号，卡片isdelete不能为true，且用户是看板所有人或看板成员，评论直接删除。