**SVN项目管理概要设计**

# 一、创建项目

需求说明：用户创建项目，同时为该项目添加成员，并指定成员的角色，为成员分配ｓｖｎ权限，并发消息告诉该成员

## 接口定义：

### 1、创建项目接口说明

请求url: proj/createProject

请求方式：post

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| name | String | 项目名称（非空） |
| code | String | 项目编号（非空、唯一） |
| startTime | Data | 开始时间 |
| endTime | Data | 结束时间 |
| note | String | 项目介绍 |
|  |  |  |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| Data | Project | Project |

返回示例：

{

"rtnCode": "1",

"rtnMsg": "1",

"data": {

"id": "2",

"name": "项目名称",

“code”:”项目编码”

"svnUrl": "svn路径",

"type": "",

"author": "1",

"departmentId": "",

"state": "1",默认1

"note": "项目说明",

“start Time”:1494482703029,

“endTime”: 1494482703029,

"createTime": 1494482703029,

"updateTime": 1494482703029,

"spare1": null,

"spare2": null,

"spare3": null

}

}

### 2、创建项目的同时添加成员？？？？？

请求url: proj/createProjectAndMember

请求方式：post

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| Name | String | 项目名称 |
| Code | String | 项目编号 |
| startTime | Data | 开始时间 |
| endTime | Data | 结束时间 |
| Note | String | 项目介绍 |
| userIds | String | 成员列表，成员之间按“，”分割 |
| 权限暂时没有设计 |  |  |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| Data | Boolean | true表示成功 |

返回示例：

### 3．根据id查询项目

请求url: proj/ getProjectById

请求方式：get

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| Id | String | 项目的id |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| Data | Project | Project |

返回示例：

{

"rtnCode": "1",

"rtnMsg": "1",

"data": {

"id": "2",

"name": "项目名称",

“code”:”项目编码”

"svnUrl": "",

"type": "",

"author": "1",

"departmentId": "",

"state": "1",默认1

"note": "项目说明",

“start Time”:1494482703029,

“endTime”: 1494482703029,

"createTime": 1494482703029,

"updateTime": 1494482703029,

"spare1": null,

"spare2": null,

"spare3": null

}

}

### 4.根据用户的id查询该用户创建的项目

请求url: proj/getProjectAllByAuthor

请求方式：get

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| author | String | 项目创建人id |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | List<Project> | Project |

返回示例：

{

"rtnCode": "1",

"rtnMsg": "1",

"data": {

"id": "2",

"name": "项目名称",

“code”:”项目编码”

"svnUrl": "",

"type": "",

"author": "1",

"departmentId": "",

"state": "1",默认1

"note": "项目说明",

“start Time”:1494482703029,

“endTime”: 1494482703029,

"createTime": 1494482703029,

"updateTime": 1494482703029,

"spare1": null,

"spare2": null,

"spare3": null

},

……

]

}

### 5. 根据id删除项目

说明：根据id删除项目,并非物理删除，而是把项目的状态改为删除

请求url: proj/delProject

请求方式：post

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| id | String | 项目id |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | boolean | true删除成功 |

返回示例：

{

"rtnCode": "1",

"rtnMsg": "3", /\*\*3表示删除成功\*/

"data": true

}

### 6. 修改项目信息

请求url: proj/updateProject

请求方式：post

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| name | String | 项目名称 |
| code | String | 项目编号 |
| startTime | Data | 开始时间 |
| endTime | Data | 结束时间 |
| note | String | 项目介绍 |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | Project | 更新成功，返回project对象数据 |

返回示例：

{

"rtnCode": "1",

"rtnMsg": "1",

"data": {

"id": "2",

"name": "项目名称",

“code”:”项目编码”

"svnUrl": "",

"type": "",

"author": "1",

"departmentId": "",

"state": "1",默认1

"note": "项目说明",

“start Time”:1494482703029,

“endTime”: 1494482703029,

"createTime": 1494482703029,

"updateTime": 1494482703029,

"spare1": null,

"spare2": null,

"spare3": null

}

}

### 7.查询所有的项目

请求url: proj/getAllProject

请求方式：get

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| 空 | s |  |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | List<Project> | Project |

返回示例：

{

"rtnCode": "1",

"rtnMsg": "1",

"data": {

"id": "2",

"name": "项目名称",

“code”:”项目编码”

"svnUrl": "",

"type": "",

"author": "1",

"departmentId": "",

"state": "1",默认1

"note": "项目说明",

“start Time”:1494482703029,

“endTime”: 1494482703029,

"createTime": 1494482703029,

"updateTime": 1494482703029,

"spare1": null,

"spare2": null,

"spare3": null

},

……

]

}

### 8.根据项目编号查询项目

请求url: proj/findByCode

请求方式：get

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| code | String | 项目编号（非空） |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | Project | Project |

返回示例：

{

"rtnCode": "1",

"rtnMsg": "1",

"data": {

"id": "2",

"name": "项目名称",

“code”:”项目编码”

"svnUrl": "",

"type": "",

"author": "1",

"departmentId": "",

"state": "1",默认1

"note": "项目说明",

“start Time”:1494482703029,

“endTime”: 1494482703029,

"createTime": 1494482703029,

"updateTime": 1494482703029,

"spare1": null,

"spare2": null,

"spare3": null

}

}

## 数据结构：

项目表PROJECT表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 说明 | 备注 |
| Id | Varchar(64) | 项目id |  |
| name | Varcahr(50) | 项目名称 |  |
| code | Varchar（64） | 项目编号 |  |
| svn\_url | Varcahr(100) | 项目svn路径 |  |
| type | Char(2) | 项目类型 |  |
| author | Var char(64) | 项目创建人id |  |
| department\_id | Var char(64) | 项目所属部门 |  |
| state | Char(2) | 状态 | 进行中、已结束 |
| note | Varchar(200) | 项目说明 |  |
| create-time | Date | 创建时间 |  |
| update\_time | Date | 更新时间 |  |
| start\_time | Date | 开始时间 |  |
| end\_time | Date | 结束时间 |  |
| spare1 | Varchar(100) | 备用字段1 |  |
| spare2 | Varchar(100) | 备用字段2 |  |
| spare3 | Varchar(100) | 备用字段3 |  |

# 二、成员管理

## 接口定义：

### 1.添加成员

请求url: pm/addProjectMember

请求方式：post

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| userId | String | 用户的id |
| projectId | String | 项目的id |
| 权限参数暂时没有 |  |  |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | ProjectMember | 添加成功，返回projecMember对象数据 |

返回示例：

{

"rtnCode": "1",

"rtnMsg": "1",

"data": {

"id": "a7178955-f249-4a0d-8139-3bc716af0220",

"userId": "用户id",

"projectId": "项目id",

"position": "",

"level": "",

"upLeaderId": "",

"isSvnPower": "",

"createTime": null,

"updateTime": null,

"spare1": null,

"spare2": null,

"spare3": null

}

}

### 2.查询所有用户

说明：用于添加添加成员时选择成员，暂时查询所有的用户，后期可以按照公司、部门等条件来查询

请求url: user/getAllUser

请求方式：get

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| 空 | 空 |  |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | List<User> |  |

返回示例：

{

"rtnCode": "1",

"rtnMsg": "1",

"data": [

{

"id": "1",

"name": "测试",

"password": "123456",

"companyId": "",

"departmentId": "",

"state": "",

"email": "106506100@qq.com",

"phone": "",

"headImg": "11111.jpg",

"svnName": "11111",

"createTime": 1494491404000,

"updateTime": 1494491450000

},

… …

]

}

### 3.删除成员

请求url: pm/delProjectMember

请求方式：post

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| Id | String | 成员id |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | Boolean | 删除成功返回true |

返回示例

{

"rtnCode": "1",

"rtnMsg": "1",

"data": true

}

### 4. 根据项目的id，查询项目下的所有成员

请求url: pm/ getMemberAllByProject

请求方式：get

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| projectId | String | 项目的id |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | List<map>  Map中包含两个对象   1. user类型对象 2. project Member类型对象 |  |

返回示例

?????

### 5 批量保存成员

请求url: pm/ saveMemberList

请求方式：get

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| projectId | String | 项目的id |
| userIds | String | 用户的id，多个用户用逗号隔开 |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | List<User> | 保存失败的用户，spare表示错误信息 |

返回示例

{

"rtnCode": "1",

"rtnMsg": "添加成功",

"data": [

{

"id": "1",

"name": "李传浩",

"password": "123456",

"company": null,

"department": null,

"state": "1",

"email": "123456@qq.om",

"phone": "18254105833",

"headImg": "qq.jpg",

"svnName": "lichh",

"createTime": 1495180206000,

"updateTime": 1495180283000,

"spare": null,

"errMsg": "该成员已添加"

}….

]

## 数据结构：

项目成员PROJECT\_MEMBER表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 说明 | 备注 |
| Id | Varchar | Id |  |
| user\_id | Varcahr | 成员id |  |
| project\_id | Varchar | 项目id |  |
| position | char | 项目职则 | 0:开发，1:UI，2:测试 |
| Level | Char | 级别 | 负责人，组长，小兵 |
| up\_leader\_id | Var char | 上级领导id |  |
| Is\_svn\_power | Char | 是否开通svn权限 | 0否，1是 |
| create \_time | Date | 创建时间 |  |
| update\_time | Date | 更新时间 |  |
| Spare1 | Var char | 备用字段1 |  |
| Spare2 | Var char | 备用字段2 |  |
| Spare3 | Var char | 备用字段3 |  |

# 3、任务分配和管理

需求说明：

用户可创建新的任务，创建完成后可将该任务分配给该项目下的某个成员，并发消息通知该成员，成员在完成该任务后需要关闭该任务

## 接口定义：

### 1．创建任务

请求url: plan/createPlan

请求方式：post

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| projectId | String | 项目id |
| name |  | 任务名称 |
| description |  | 任务描述 |
| owner |  | 任务拥有者id，默认为创建人 |
| solver |  | 任务解决着者id |
| state |  | 状态 |
| startTime |  | 开始时间 |
| endTime |  | 结束时间 |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | 任务对象类型 | 任务对象 |

返回示例：

{

"rtnCode": "1",

"rtnMsg": "1",

"data": {"id": "d7cd7c5e-106b-4709-a65f-0535fefc2479",

"projectId": "项目id",

"name": "任务名称",

"description": "任务描述",

"createrId": "创建人id",

"owner": "任务拥有者id",

"solver": "任务解决着者id",

"state": "状态",

"level": "优先级",

"startTime": 开始时间,

"endTime": 结束时间,

"createTime": 创建时间,

"updateTime": 更新是假,

"spare1": 备用字段1,

"spare2": 备用字段2

}

}

### 2．根据任务id查询任务及任务的操作记录

请求url: plan/getPlanById

请求方式：get

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| id | String | 任务id |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | plan | 任务对象 |

返回示例：

{

"rtnCode": "1",

"rtnMsg": "查询成功",

"data": {

"list": [

{

"id": 3,

"businessId": 1,

"operatorId": 1,

"createrId": 1,

"createTime": "14:38:18",

"endTime": "14:38:18",

"operationType": "",

"description": "编辑任务"

},

……

],

"plan": {

"id": 1,

"projectId": null,

"name": "12345",

"description": "1111111",

"createrId": null,

"owner": "1",

"solver": "1",

"state": "0",

"level": null,

"startTime": null,

"endTime": null,

"createTime": 1495538783000,

"updateTime": 1495538783000,

"spare1": null,

"spare2": null

}

}

}

### 3．根据项目的id查询该项目下的所有任务

请求url: plan/getPlanAllByProjectId

请求方式：get

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| Id | String | 项目的id |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | List<plan> | 任务对象列表 |

返回示例：

{

"rtnCode": "1",

"rtnMsg": "1",

"data": [

"data": {

"id": "d7cd7c5e-106b-4709-a65f-0535fefc2479",

"projectId": "项目id",

"name": "任务名称",

"description": "任务描述",

"createrId": "创建人id",

"owner": "任务拥有者id",

"solver": "任务解决着者id",

"state": "状态",

"level": "优先级",

"startTime": 开始时间,

"endTime": 结束时间,

"createTime": 创建时间,

"updateTime": 更新是假,

"spare1": 备用字段1,

"spare2": 备用字段2

}，

…...

]

}

### 4．更新任务信息

说明：更新任务信息：包括状态信息、关闭任务、删除任务、分配任务等操作

请求url: plan/ updatePlan

请求方式：post

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| projectId | String | 项目id |
| name |  | 任务名称 |
| description |  | 任务描述 |
| owner |  | 任务拥有者id,默认为创建人 |
| solver |  | 任务解决着者id |
| state |  | 状态 |
| startTime |  | 开始时间 |
| endTime |  | 结束时间 |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | Plan | 任务对象 |

返回示例：

{

"rtnCode": "1",

"rtnMsg": "1",

"data": {

"id": "d7cd7c5e-106b-4709-a65f-0535fefc2479",

"projectId": "项目id",

"name": "任务名称",

"description": "任务描述",

"createrId": "创建人id",

"owner": "任务拥有者id",

"solver": "任务解决着者id",

"state": "状态",

"level": "优先级",

"startTime": 开始时间,

"endTime": 结束时间,

"createTime": 创建时间,

"updateTime": 更新是假,

"spare1": 备用字段1,

"spare2": 备用字段2

}

}

### 5．根据id查询任务的操作记录

请求url: po/getPlanOperationById

请求方式：get

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| Id | String | 任务id |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | PlanOperation | 任务操作记录 |

返回示例：

### 6．根据任务id，查询该任务的所有操作记录

请求url: po/ getPlanOperationAllByPlanId

请求方式：get

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| Id | String | 任务id |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | List<PlanOperation> | 任务操作记录列表 |

返回示例：

### 7．查询当前用户所有未处理的任务

请求url: plan/ findPlanByOperator

请求方式：get

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| userId | String | 用户的id |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | List<Plan> | 任务列表 |

返回示例：

### 8．查询用户的所有已分配或者已解决待验证的任务

请求url: plan/ findPlanByCreater

请求方式：get

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
| userId | String | 用户的id |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | List<Plan> | 任务列表 |

返回示例：

{

"rtnCode": "1",

"rtnMsg": "查询成功",

"data": [

{

"id": 3,

"projectId": null,

"name": "lichuanhao",

"description": "123456789",

"createrId": "1",

"owner": "1",

"solver": "",

"state": "0",

"level": null,

"startTime": null,

"endTime": null,

"createTime": 1495541834000,

"updateTime": 1495541834000,

"spare1": null,

"spare2": null

},

…..

]

}

数据结构：

任务表PLAN表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 说明 | 备注 |
| Id | varchar(64) | Id |  |
| Project\_id | varchar(64) | 项目id |  |
| Name | varchar(200) | 任务名称 |  |
| description | varchar(500) | 任务描述 |  |
| Creater\_id | varchar(64) | 创建者id |  |
| Owner | varchar(64) | 任务拥有者 |  |
| Solver | varchar(64) | 任务的解决者 |  |
| State | Char(2) | 任务状态 | 0未解决、1已解决待验证、2已解决 |
| Level | Char(2) | 任务的优先级 |  |
| solution | Varchar | 解决方案 |  |
| workCount | float | 工作量 |  |
| Carate\_time | Date | 创建时间 |  |
| Update\_time | Date | 更新时间 |  |
| Start\_time | Date | 开始时间 |  |
| End\_time | Date | 结束时间 |  |

任务操作记录表TASK\_INST表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 说明 | 备注 |
| Id | varchar(64) | Id |  |
| business\_id | varchar(64) | 业务id |  |
| operator \_id | varchar(64) | 执行人id |  |
| creater\_id | Varchar | 创建人id |  |
| create\_time | date | 创建时间 |  |
| end\_time | date | 结束时间 |  |
| operation\_type | char(2) | 操作类型 | 创建、关闭任务、激活任务、转发任务等 |
| description | varchar(200) | 操作描述 |  |

# 4、用户管理

需求说明：

暂时自己建几张表表示用户的信息，后期用户的信息会从框架带的服务中获取

## 接口定义：

### 1．查询所有用户

请求url: user/findAllUser

请求方式：get

请求参数：

|  |  |  |
| --- | --- | --- |
| **入参** | **参数类型** | **参数说明** |
|  |  |  |

返回参数：

|  |  |  |
| --- | --- | --- |
| **返回参数** | **参数类型** | **参数说明** |
| rtnCode | String | 0表示存在错误，1表示返回正确 |
| rtnMsg | String | 当rtnCode=o时，rntMsg表示错误信息  当rtnCode=1时，rntMsg表示正确的信息  （还没有定义详细的标准，后期会完善起来） |
| data | 任务对象类型 | 任务对象 |

返回示例：

数据结构：

用户USER表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 说明 | 备注 |
| Id | varchar(64) | Id |  |
| name | varchar(64) | 用户名 |  |
| password | varchar(64) | 密码 |  |
| company\_id | varchar(64) | 公司id | 外键 |
| department\_id | varchar(64) | 部门id | 外键 |
| state | varchar(2) | 状态（？？？） |  |
| email | varchar(64) | 邮箱 |  |
| phone | varChar(64) | 电话 |  |
| head\_img | varchar(100) | 头像 |  |
| svn\_name | Varchar(30) | Svn用户名 |  |
| Carate\_time | Date | 创建时间 |  |
| Update\_time | Date | 更新时间 |  |

部门表department

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 说明 | 备注 |
| Id | varchar(64) | Id |  |
| name | varchar(64) | 部门名称 |  |
| company\_id | varchar(64) | 公司id | 外键 |
| note | varchar(200) | 部门介绍 |  |
| leader\_id | varchar(64) | 领导人id | 外键 |
| state | varchar(2) | 状态（？？？） |  |
| up\_department\_id | Var char(64) | 上级部门id |  |
| level | Var char(2) | 部门级别（0、1、2、3…如0表示没有上级部门，1表示有上级部门并处在第二层） |  |
| create\_time | Date | 创建时间 |  |
| update\_time | Date | 更新时间 |  |

公司表company

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 说明 | 备注 |
| Id | varchar(64) | Id |  |
| name | varchar(64) | 公司名称 |  |
| note | varchar(200) | 公司介绍 |  |
| leader\_id | varchar(64) | 领导人id | 外键（多对一） |
| state | varchar(2) | 状态（？？？） |  |
| create\_time | Date | 创建时间 |  |
| update\_time | Date | 更新时间 |  |