# Model层API

## user

|  |  |  |  |
| --- | --- | --- | --- |
| id | PK auto increment | INTEGER |  |
| username | Not null | STRING(10) |  |
| password | Not null | STRING(40) | 存加密后的字符串 |
| avatar |  | STRING(40) | 存储头像路径 |
| phone |  | STRING(11) |  |
| email |  | STRING(40) |  |
| idcard |  | STRING(18) |  |
| idcardfrontpic |  | STRING(40) | 存储身份证正面照路径 |
| idcardbackpic |  | STRING(40) | 存储身份证背面照路径 |
| identity | Default 0 | CHAR(1) | 状态 正常1 多的另加 |
| credit | Default 700 | INTEGER | 信用评分 致敬芝麻信用 |
| tags |  | STRING(20) | 标签 1&2&3的形式 |
| school |  | STRING(20) | 学生老师就school  员工就company |
| discipline |  | STRING(20) | 学生就专业  员工就部门 |
| experience |  | STRING(100) | 工作经历 100字以内 |
| introduction |  | STRING(100) | 简介 100字以内 |

## requirement

|  |  |  |  |
| --- | --- | --- | --- |
| id | PK auto increment | INTEGER |  |
| ownerId | Not null | INTEGER | 需求提出人的ID |
| ownerDesc | Not null | STRING(20) | 需求提出人的描述例 清华大学软件学院某某教授 |
| avatar | Not null | STRING(40) | 需求图片存储路径 例 清华大学老师就直接放logo好了 |
| title | Not null | STRING(20) | 需求标题 |
| tags | Not null | STRING(20) | 需求标签 1&2&3的形式 |
| content | Not null | STRING(100) | 需求内容 |
| city |  | STRING(20) | 所在城市 |
| type |  | STRING(20) | 项目类别 |
| discipline |  | STRING(20) | 专业 |
| acceptorId |  | INTEGER | 需求接受人的ID |
| state |  | CHAR(1) | 状态 具体再细化 |

## friend

|  |  |  |  |
| --- | --- | --- | --- |
| id | PK auto increment | INTEGER |  |
| userId1 | Not null | INTEGER | 友人A的id 注意友人的身份是双向的 所以存入好友信息时 存入AB的同时也得存入BA 删除也得同时 |
| userId2 | Not null | INTEGER | 友人B的id |

## tag

|  |  |  |  |
| --- | --- | --- | --- |
| id | PK auto increment | INTEGER |  |
| name | Not null | STRING(10) |  |

# Service层API

## userService

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Input | Output | Desc |
| createUser | user对象 包含如下属性  user.username, user.password, user.avatar, user.phone, user.email, user.idcard, user.idcardfrontpic, user.idcardbackpic, user.identity, user.state, user.credit, user.tags, user.school, user.discipline, user.experience, user.introduction | 完整的user对象 | 创建某用户信息  列出的属性必须填满 没有填写的话设成'' |
| updateUser | newValues, id  newValues是Object对象 需要在Controller层打包好直接传过来  比如更新了phone信息和email信息  传过来的newValues = { phone : '18801180000', email : 'foobar@126.com' }  id是where查询条件 用户的ID | 完整的user对象 | 补全某用户信息 |
| findUser | need,query  need是需要获得的属性  比如登录需要 id state和password  need = ['id','state','password']  query是where查询条件  比如检查用户名是否存在  query = { username : 'foobar' } | user对象  只有need里包含的属性 | 查找某用户  如登录时用到 |
| findUsers | need,query  同上 | user对象list  通过索引获取单独的某个用户 | 查找多用户 |
| isUserExisted | name  要输入的用户名 | 返回true或false | 检查用户名是否存在 |
| findTags | id 用户的id | 返回list Object.name是标签的名字 | 根据用户id返回用户的标签列表 |
| addTag | id 用户的id  tag 要新加的标签名字 | 标签已存在或其他错误 返回false  否则返回user对象 | 为用户添加标签 |
| authUser | name,password | user对象 | 认证用户  还未实现 |
| findUserAll | id | 完整user对象 | 根据id查找user所有属性 |

## requirementService

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Input | Output | Desc |
| createRequirement | requirement对象  包含以下属性  requirement.ownerId, requirement.ownerDesc, requirement.avatar, requirement.title, requirement.tag, requirement.content,  requirement.city,  requirement.type,  requirement.discipline, requirement.acceptorId, requirement.state | 完整的requirement对象 | 发布需求  列出的属性必须填满 没有填写的话设成'' |
| updateRequirement | newValues,id  newValues是Object对象 需要在Controller层打包好直接传过来  比如更新了content信息和acceptorId信息 传过来的newValues = { content : 'new content', acceptorId : 123 }  id是where查询条件 需求的ID | 完整的更新后的requirement对象 | 更新需求 如接受需求 解雇承包人等 |
| findRequirement | need, query  例：need=[‘content’,’id’,’title’]  query={title:‘123’} | 某个requirement对象 | 查找单个需求 |
| findRequirements | need, query  同上 | 多个符合条件requirement对象的list | 查找多个需求 |
| findByCity | city 所在城市 | 多个符合条件requirement对象的list | 根据城市查找需求 |
| findByType | type 项目类别 | 多个符合条件requirement对象的list | 根据项目类别查找需求 |
| findByDiscipline | discipline 专业 | 多个符合条件requirement对象的list | 根据专业查找需求 |
| findTags | id  需求的id | 返回list  Object.name是标签的名字 | 返回需求的标签 |
| addTag | id,tag  id需求的id  tag 要新加的标签名字 | 标签已存在或其他错误 返回false  否则返回requirement对象 | 给需求添加标签 |

## friendService

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Input | Output | Desc |
| createFriend | id1,id2  id1和id2是两个用户的id | 返回user对象 | 双向添加好友 |
| deleteFriend | id1,id2  id1 和id2是两个用户的id | 返回user对象 | 双向删除好友 |
| findFriends | id1 | 返回Array对象 里面只有好友的id | 获取用户的好友列表 |
| findFriendsInfo | id1 | 返回List 里面包含了多个用户  每个用户包含了  id username avatar phone  email identity  school discipline信息 | 获取所有好友的基本信息 用于展示人脉列表 |
| isFriend | id1,id2 | 返回true或false | 判断用户id1是否有用户id2这个朋友 |