

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
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	user 对象 list	查找多用户

	同上	通过索引获取单独的某个用户	
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.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
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