软件工程小项目运行流程

目录

[1.登录 2](#_Toc21133602)

[1.1已有账户，直接登录 2](#_Toc21133603)

[1.2注册用户 2](#_Toc21133604)

[2.查看组队学习信息 3](#_Toc21133605)

[2.1第一次进入该界面，分页查询 3](#_Toc21133606)

[2.2根据选择的查询条件查询数据，分页查询 4](#_Toc21133607)

[2.3申请入队 4](#_Toc21133608)

[3.查看玩乐信息 5](#_Toc21133609)

[3.1第一次进入该界面，分页查询 5](#_Toc21133610)

[3.2根据选择的查询条件查询数据，分页查询 5](#_Toc21133611)

[3.3申请入队 6](#_Toc21133612)

[4.个人主页 6](#_Toc21133613)

[4.1显示个人信息 6](#_Toc21133614)

[4.2显示用户发布过的组队信息 7](#_Toc21133615)

[4.3查看用户参加过的组队信息 7](#_Toc21133616)

## 1.登录

### 1.1已有账户，直接登录

1.前台：

获取用户输入的‘用户名{username}’与‘密码{password}’

2.调用后台接口：

‘UserInfoController’下‘login’接口，传入‘user{username,password}’对象。

返回值：使用数字代表不同的状态：

1：登录成功

2：该用户名未注册

3：密码错误

### 1.2注册用户

1.前台：

获取用户的基本信息：

1.用户名{username}【必填】

2.密码{password}【必填】

3.昵称{nickname}【必填】

4.性别{sex}【必填】

5.所在城市{city}【必填】

6.所在学校{school}【必填】

7.所在院系{academy}

8.所在专业{major}

9.个性签名{personalized\_signature}

10.个人标签{personalized\_labels}

前台给用户设置默认的头像，修改用户头像在用户的个人主页实现，注册界面不添加此功能，该头像不传入后台。

2.调用后台接口：

‘UserInfoController’下‘insert’接口，传入 ‘userInfo’ 对象：

userInfo：{username，password，nickname，sex，city，school，academy，major，personalized\_signature，personalized\_labels}

返回值：使用数字表示不同的状态

1：注册成功

2：该昵称已被使用

3：该用户名已被使用

## 2.查看组队学习信息

### 2.1第一次进入该界面，分页查询

1.前台：

显示从后台拿到的数据。

2.调用后台接口：

‘StudyInfoController’下‘getStudyInfo’接口。传入‘username’参数，返回一个‘PageBean<StudyInfo>’给前台

PageBean：{

total：总条数

list：返回的结果集，List<StudyInfo>

pageNum：当前页

pages：总页数

pageSize：每一页的结果集的数量

size：当前页的结果集的数量

}

StudyInfo:{

id：id值

username：用户名

userImgUrl：用户头像地址

userSchool：用户所在的学校

type：模式，1：线下，2：线上

maxMember：最大组队人数，包括用户

leftMember：剩余空位

label：标签

time：学习时间

site：学习地点

ps：备注

date：发布时间

valid：是否有效

}

3.后台代码实现：

根据传入的‘username’获取该用户所在的学校‘school’，根据模式的不同，各提取10条最新的数据。注意查询条件

1.查询方式为分页查询

2.查询线下模式时，表中school字段应与用户所在学校一致。

3.查询到的数据要为最新的

拿到数据之后，补全给条信息中，用户头像的地址，并将数据进行乱序处理，返回一个‘PageBean<StudyInfo>’给前台

### 2.2根据选择的查询条件查询数据，分页查询

1.前台：

提供选择框，保存用户选择的条件，

1：type，模式，【使用数字代替模式，1：线下，2：线上，必选】

2：label，学习标签，【可不选，选择只能是单选】

3：maxMember，组队人数，【可不选，也可指定组队人数，最多为6人。】

2.调用后台接口：

调用‘StudyInfoController’下‘selectStudyInfo’，传入‘queryInfo’对象【QueryStudyInfo{type，username，label，maxMember}】，‘pageNum’【当前页】，‘pageSize’【当前页显示的数据量】，返回一个‘PageBean<StudyInfo>’给前台。

3.后台代码实现：

后台根据传入的条件进行分页查询，返回一个‘PageBean<StudyInfo>’给前台。注意查询条件：

1.线下模式需要获取当前用户所在的学校，查询到的数据中的‘school’字段的值与该用户所在的学校一致。

2.查询到的结果应该倒序排列，因为向用户显示的数据顺序应该是最新插入的数据排在最前

3.使用分页操作对查询到的数据进行处理。

### 2.3申请入队

1.前台：

点击按钮申请组队，注意，不可申请无效的组队。前台注意，对于无效的组队信息，设置按钮无效或者不显示。

2.调用后台接口：

调用后台‘auditInfoController’下‘insert’接口，传入‘AuditInfo’对象。

AuditInfo：{

sender：申请人，填入当前用户名

receiver：接收者，填入选中的组队信息的发布人

actionType：活动类型，使用数字代替，1：线下学习，2：线上学习，3：吃喝玩乐

actionId：活动id，获取该活动的id

auditStatus：申请状态，填入0

auditResult：申请结果，填入0

}

3.后台代码实现：

1.向auditInfo表中插入一条数据

## 3.查看玩乐信息

### 3.1第一次进入该界面，分页查询

1.前台：

显示从后台拿到的数据

2.调用后台接口

调用‘enjoyInfoController’下‘getEnjoyInfo’接口，传入当前用户的‘username’，返回一个‘PageBean<EnjoyInfo>’数组，长度为20条

EnjoyInfo：{

`id`：主键,  
`username`：用户名,

`userImgUrl`：用户头像地址  
`user\_city`：用户所在城市,  
`label`：标签,  
`sex\_specific` 性别限制，性别限制分三种，用数字代替，1：只限异性，2：只限同性，2：无限制  
`time`：玩乐时间,  
`site`：玩乐地点

`max\_member`：最大组队人数

`left\_member`：剩余空位  
`ps`：备注

`date`：消息发布的时间

`valid`：消息是否有效

}

3.后台代码实现：

注意查询条件：

1.用户只能查询到与自己城市一致的人发布的玩乐信息。

2.查询20条最新的数据

3.分页查询

### 3.2根据选择的查询条件查询数据，分页查询

1.前台：

提供选择框，保存用户选择的条件，

1：label，玩乐标签，【可不选，选择只能是单选】

2：maxMember，组队人数，【可不选，也可指定组队人数，最多为6人。】

3：sexSpecific，性别限制，用数字标注，1：只限异性，2：只限同性，3：无限制。

2.调用后台接口：

调用‘EnjoyInfoController’下‘selectEnjoyInfo’，传入‘queryInfo’对象【QueryStudyInfo{username，label，maxMember，sexSpecific }】，‘pageNum’【当前页】，‘pageSize’【当前页显示的数据量】，返回一个‘PageBean<EnjoyInfo>’对象

3.后台代码实现：

后台根据传入的条件进行分页查询，返回一个‘PageBean<EnjoyInfo>’给前台。注意查询条件：

1.查询到的数据中的城市（city字段）要与用户所在的城市一致

2.查询到的结果应该倒序排列，因为向用户显示的数据应该是最新插入的数据

3.使用分页操作对查询到的数据进行处理。

### 3.3申请入队

1.前台：

点击按钮申请组队，注意，不可申请无效的组队。前台注意，对于无效的组队信息，设置按钮无效或者不显示。

2.调用后台接口：

调用后台‘auditInfoController’下‘insert’接口，传入‘AuditInfo’对象。

AuditInfo：{

sender：申请人，填入当前用户名

receiver：接收者，填入选中的组队信息的发布人

actionType：活动类型，使用数字代替，1：线下学习，2：线上学习，3：吃喝玩乐

actionId：活动id，获取该活动的id

auditStatus：申请状态，填入0

auditResult：申请结果，填入0

}

## 4.个人主页

### 4.1显示个人信息

1.前台：

显示个人信息，注意界面的布局

2.调用后台接口：

调用‘UserInfoController’下‘getUserInfo’接口，传入参数‘username’，后台返回一个‘UserInfo’对象。

userInfo：{

username：用户名，

nickname：昵称，

sex：性别，

city：所在城市，

school：所在学校，

academy：所在院系，

major：所学专业，

personalized\_signature：个性签名，

personalized\_labels：个人标签

}

3.后台逻辑实现：

1.补全用户头像的地址，方便前台做数据渲染

2.使返回给前台的 ‘password’为null值

### 4.2显示用户发布过的组队信息

1.前台

显示从后台拿到的数据。注意数据渲染，未失效的组队信息与失效的组队信息的显示应该要有明显的区别。

2.调用后台接口：

1.查询用户发布过的组队学习信息

调用‘StudyInfoController’下‘getStudyInfoByUserIssue’接口，传入用户名，返回一个‘PageBean<StudyInfo>’对象。

2.查询用户发布过的玩乐信息

调用‘EnjoyController’下‘getEnjoyInfoByUserIssue’接口，传入用户名，返回一个‘PageBean<EnjoyInfo>’对象。

3.后台代码实现：

1.分页查询

2.后台对数据进行排序处理，将插到的消息分为‘未失效’，‘失效’两类，‘未失效的信息’排在‘失效信息’前面，对于各类信息，按照时间‘由近及远’的顺序进行排序。

### 4.3查看用户参加过的组队信息

1.前台

显示从后台拿到的数据。注意数据渲染，未失效的组队信息与失效的组队信息的显示应该要有明显的区别。

2.调用后台接口：

1.查询用户参与过的组队学习信息

调用‘StudyInfoController’下‘getStudyInfoByUserAttend’接口，传入参数：用户名【username】，返回一个‘PageBean<StudyInfo>’对象。

2.查询用户发布过的玩乐信息

调用‘EnjoyController’下‘getEnjoyInfoByUserAttend’接口，传入参数：用户名【username】，返回一个‘PageBean<EnjoyInfo>’对象。

3.后台代码实现：

1.分页查询，分页用于查询‘team\_info’表

2.多表查询，先‘team\_info’表中根据‘username’，‘action\_type’查询对应的‘action\_id’，在根据‘action\_id’拿到对应的数据。