

《Java 语言程序设计》实验报告

实验三 基于触发器-动作框架的在线 IFTTT 系统

131220142 江悦
131220151 潘秋红
131220169 汪倩雯

一、程序整体结构

后台共分为四个模块,即四个包,分别是 database, servlet, task, user 包。database 包主要负责数据库的连接、关闭,增删减查用户数据的功能;servlet 包则是各种 servlet 的集合,用于响应前台;task 包包装了与 ifttt 任务相关的操作;user 包里封装了用户数据。

二、各模块的功能与接口

1、user package

定义了一个 User 类,如下:

```
public class User {  
    private String username;  
    private String password;  
    private String sex;  
    private String email;  
    private int level;  
    private int account;  
    .....  
}
```

2、task package

1) Task

是存储任务信息的类,属性有用户名、任务的 this 和 that 事件、事件的基本信息如用户的账号密码、任务的状态等。

2) TaskMap

用 HashMap 存储任务运行中的一些实时数据。比如 key 是任务 ID, value 是任务状态,可以较快获得任务的状态。

<code>void prepareStateMap()</code>	准备 StateMap,初始状态都是未开始;在 TaskManage.jsp 中调用
<code>void setState(int id, int state)</code>	设置任务状态,

	StartTaskServlet 和 StopTaskServlet 会调用此方法。
<code>int getState(int taskId)</code>	获得任务状态，在 StartTaskServlet 和 StopTaskServlet 调用。
<code>Timer getTimer(int taskId)</code>	获得某一任务的 Timer
<code>TimerTask getTimerTask(int taskId)</code>	获得某一任务的 TimerTask，在 StopTaskServlet 中调用。
<code>setTimerTask(int id, TimerTask tt)</code>	设置某任务的 TimerTask，在 StartTaskServlet 中调用。

3) CheckWeibo

继承了 TimerTask 类，重写了 `public void run()` 方法。负责监听指定 ID 的微博，当其发布含有指定内容的微博时，触发 that 事件。与 Timer 一同工作，每 10 秒执行一次。

在 `run()` 方法中，先获得微博授权，再获得当前用户的微博，判断是否有包含指定内容的新微博。

在 StartTaskServlet 中调用此类。

4) MonitorWeibo

同样继承 TimerTask 类。也是监听指定 ID 的新浪微博，如果在指定时间内未发布任何新微博，则触发 that 事件。与 Timer 一起工作，延迟一段时间后开始执行。

在 `run()` 方法中，同样要先得到微博的授权，获得用户的微博列表，如果没有在开始时间后发布的微博，就满足了 this 事件，执行对应的 that 事件。

在 StartTaskServlet 中调用此类。

5) CheckMail

继承 TimerTask 类，实现当 This 任务是判断指定邮箱是否收到邮件时的任务执行。在 `run()` 方法中，先是根据用户设置的 QQ 邮箱用户名和密码获取链接，然后判断当前收件箱的邮件数是否比上一次的多，如果变多了，则说明收到了邮件，执行 That 任务：当 That 任务是发送邮件时，创建一个 SendMail 类型的对象，调用 `void task.SendMail.run()` 方法实现发送邮件。如果 That 任务是发送微博，则创建一个 SendWeibo 类型的对象，调用 `void task.SendWeibo.run()` 方法实现发送微博。

6) SendMail

继承 `TimerTask` 类，实现发送邮件的功能。在 `run()` 方法中，设置好邮件服务器并获取默认 `session` 对象后，通过 `id,password,toAddrs` 和 `content` 属性设置 `MimeMessage`，然后将邮件发送。

7) SendWeibo

继承 `TimerTask` 类，实现发送微博的功能。在 `run()` 方法中，根据 `token` 属性的值创建 `Timeline` 类型的对象，通过 `Status` `weibo4j.Timeline.updateStatus(String status) throws WeiboException` 方法发送微博。

3、database package

1) ConnectDB 类

主要功能是建立、关闭数据库的连接，有 `public static Connection getConnection()` 和 `public static void closeConnection(Connection con)` 两个方法。

2) DBFunction

用于实现用户数据的增删改查，根据前台的具体操作决定调用哪个方法。

<code>Task getTaskById(int taskId)</code>	通过 <code>taskId</code> 获得对应的 <code>Task</code> 对象
<code>User getUserByName(String username)</code>	通过用户名得到 <code>User</code> 对象
<code>void deleteTask(String taskId)</code>	根据 <code>taskId</code> 删除数据库中对应的 <code>task</code>
<code>void insertTask(Task task)</code>	传入一个 <code>Task</code> 对象，插入

	到数据库中。
<code>boolean</code> <code>userIsExist(String username)</code>	判断给定用户名是否已存在
<code>User</code> <code>login(String username,String password)</code>	根据用户名和密码登录，如果正确，则返回一个 <code>User</code> 对象，否则返回 <code>null</code>
<code>void</code> <code>sendNotice(String text)</code>	管理员发公告，内容是 <code>text</code> 。由 <code>SendNoticeServlet</code> 类的 <code>doPost</code> 方法调用。
<code>void</code> <code>sendPrivate(String username,String text)</code>	给用户名为 <code>username</code> 的用户发私信
<code>String</code> <code>getMessages(String username)</code> <code>throws</code> <code>Exception</code>	获得某一用户收到的公告和私信
<code>void</code> <code>insertConsum(User user,int id,String cost,String alarm)</code>	插入一条消费记录
<code>int</code> <code>numOfConsum(String username)</code>	获得某用户的消费记录的数量
<code>void</code> <code>deleteMessage(String taskid)</code>	删除私信
<code>void</code> <code>changeMessage(String content,String id)</code>	修改私信或公告

4、servlet package

1) LoginServlet

负责用户的登录和注册。当用户点击 `index.jsp` 的登录或注册按钮时，后台会执行 `LoginServlet` 的 `doPost` 方法。会调用 `DBFunction` 类中的 `login()` 方法，或直接导航到 `Signup.jsp`

2) SignupServlet

处理 `Signup.jsp` 界面的 `post` 请求，先是调用 `boolean`

`database.DBFunction.userIsExist(String username)`方法判断当前用户名是否在数据库中已存在，若存在，则跳转到 `signerror.jsp` 界面，否则，调用 `void database.DBFunction.insertUser(User user)`方法将该用户加入数据库中。

3) NewTaskServlet

实现新建任务功能，在 `LoginSuccess.jsp` 点击确定，调用此 `servlet`。先根据用户输入的信息实例化一个 `Task` 类，然后调用 `DBFunction` 的 `insertTask()`方法把任务插入到数据库中。

4) DeleteTaskServlet

实现删除任务功能。在 `TaskManage.jsp` 界面点击任务后面的垃圾箱按钮，调用此 `servlet`。调用 `DBFunction` 的 `deleteTask()`方法。

5) StartTaskServlet

开始任务。在 `TaskManage.jsp` 界面点击任务后面的电源按钮，调用此 `servlet`。先根据 `taskID` 获得一个 `Task` 对象，调用的是 `DBFunction` 的 `getTaskById()`方法。

然后调用 `User database.DBFunction.getUserByName(String username)`方法获取当前用户信息，并判断用户当前余额是否充足。余额不够时提醒用户并跳回到 `TaskManage.jsp` 界面，否则，更新该用户的等级和余额，并调用 `void database.DBFunction.insertConsum(User user, int id, String cost, String alarm)`方法将这笔消费插入消费列表数据库中。

然后根据 `Task` 的 `this` 和 `that` 事件，初始化各自的 `Timer` 和 `TimerTask`，

并把 Timer 和 TimerTask 放到 TaskMap 的 HashMap 里面，方便以后查询。

调用 TaskMap 的 setTimerTask() 和 addTimer() 方法。

6) StopTaskServlet

停止任务。在 TaskManage.jsp 界面点击任务后面的电源按钮，调用此 servlet。在 doGet() 方法中，调用 TaskMap 的 getTimer() 方法，根据 taskID 获得对应的 Timer，执行 Timer 的 cancel() 方法停止任务。

7) AuthorizeServlet

获得微博授权的 code。点击授权页面 Authorize.jsp 的确定按钮，调用此 servlet。在 doPost() 方法中，调用 TaskMap 的 getTimerTask() 方法获得给定 taskID 的 TimerTask，即 SendWeibo 或 CheckWeibo 或 MonitorWeibo 的实例，调用他们的 setCode() 方法。

8) SendNoticeServlet

实现管理员发公告功能，调用 DBFunction 类的 sendNotice() 方法。

9) ModifyTaskServlet

修改任务。在 TaskManage.jsp 界面点击任务后面的编辑按钮，调用此 servlet。在 doPost() 方法中，更新任务信息。

10) ChangeProfileServlet

修改个人信息和充值。在 UserProfile.jsp 页面点击保存或充值，调用此 servlet。在 doPost() 方法中，调用 ConnectDB 类的 getConnection() 方法，更新数据库中的用户信息。

11) SendPrivateServlet

管理员向指定用户发送私信。处理 SendPrivate.jsp 界面的 post 请求，获取私信的收件人和内容后，调用 void database.DBFunction.sendPrivate(String username, String text)方法发送，完成后调回到管理员登录后的初始见面。

12) MessageManageServlet

实现管理员对已发布的公告或私信的删除和修改功能。处理 MessageDetail.jsp 界面的 post 请求，判断当前动作是删除还是修改。当管理员选择删除消息时，调用 void database.DBFunction.deleteMessage(String id)方法，将该消息删除；当管理员选择修改消息时，调用 void database.DBFunction.changeMessage(String content, String id)方法将该消息修改。处理完成后，跳转到 MessageManage.jsp 界面显示当前所有消息。

13) SetLevelServlet

管理员设置用户等级。处理 UserDetail.jsp 界面的 post 请求，获取当前界面选中的用户和要修改的等级，修改成功后刷新当前界面。

5. 前台

Admin.jsp: 管理员登录初始界面

Authorize.jsp: 微博任务选择界面

index.jsp: 初始界面

LoginSuccess.jsp: 普通用户登录成功界面

LoginFailed.jsp: 普通用户登录失败界面

MessageDetail.jsp: 消息详情界面

MessageManage.jsp: 消息管理界面

Messages.jsp: 普通用户消息界面

ModifyTask.jsp: 修改任务界面

SendNotice.jsp: 发送公告界面

SendPrivate.jsp: 发送私信界面

signerror.jsp: 注册失败界面

signsuccess.jsp: 注册成功界面

Signup.jsp: 用户注册界面

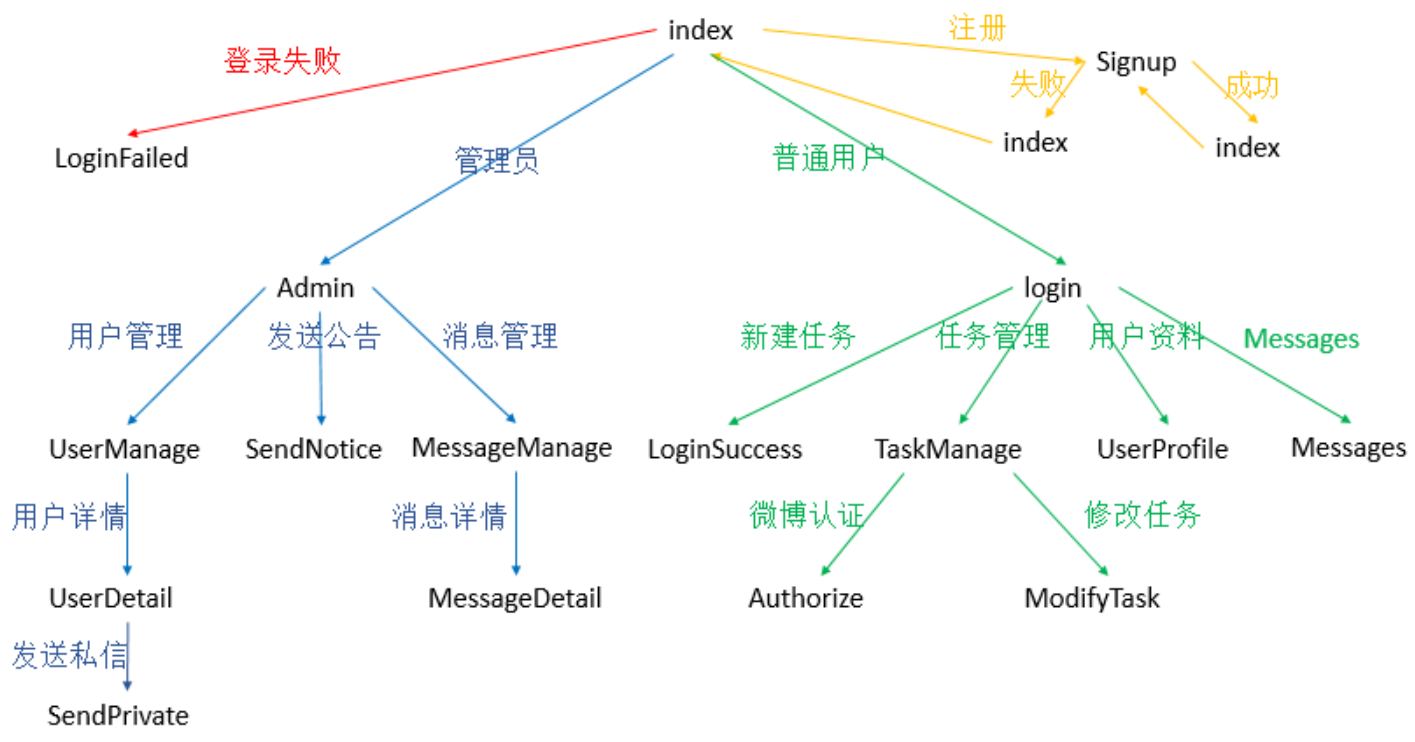
TaskManage.jsp: 任务管理界面

UserDetail.jsp: 用户详情界面

UserManage.jsp: 用户管理界面

UserProfile.jsp: 用户资料界面

各界面调用关系图：



二、数据库配置说明

使用 MySQL 数据库，执行 lab3.sql 文件中所有语句。修改 database 包的 ConnectDB.java 文件，将第 28 行的方法参数改成自己的用户名和密码。

三、成员分工

学号	姓名	分工	打分
131220142	江悦	用户登录； 新建任务； 修改任务； 开始、停止任务； 任务的具体实现：监听微博是否发送指定内容；监听微博是否在指定时间未发布微博；微博授权； 修改用户信息； 管理员发公告； 数据库连接； 部分数据库功能的实现； 部分前台界面实现；	85
131220151	潘秋红	用户注册；	85

		用户查看公告、私信 用户等级相关内容； 消费记录； 管理员查看用户详情； 管理员发私信； 管理员消息管理； 任务的具体实现：监听是否收到新邮件；发送 QQ 邮件；发送微博； 部分数据库功能的实现； 部分前台界面实现；	
131220169	汪倩雯	前台界面实现；	80