

一、项目介绍

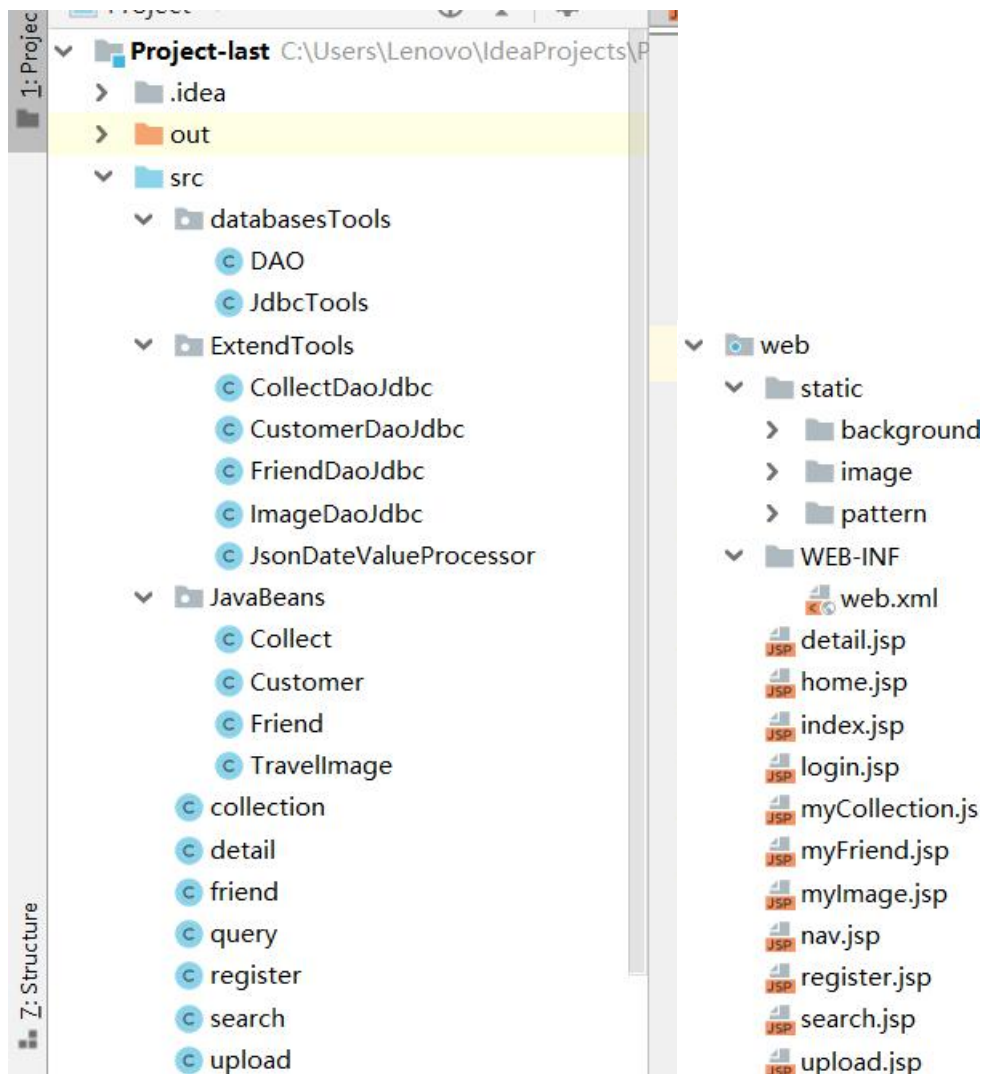
本次项目是编写一个小型旅游图片分享的网站，重点是运用 jsp+servlet+javabean 的 MVC 架构实现，前端可以使用 bootstrap 等框架，数据库使用 mysql，服务器是使用 Apache Tomcat。

二、基本功能

有两种身份访问网站，一种是未登录时，可以访问首页、搜索、查看图片细节、进行登录或注册，而登陆之后可以对图片进行收藏、上传并查看我的收藏、我的上传以及查看好友列表并查看好友的收藏，最后可以退出登录。

三、实现方式

首先看一下目录结构：



其中 javabean 包由面向对象思想，编写了四个对象类，分别表示收藏类、用户类、好友类以及图片类，databasesTools 包中 DAO 提供泛型的四种方法，分别对数据库进行增删改查功能：

```

private Class<T> clazz;
private QueryRunner queryRunner=new QueryRunner();

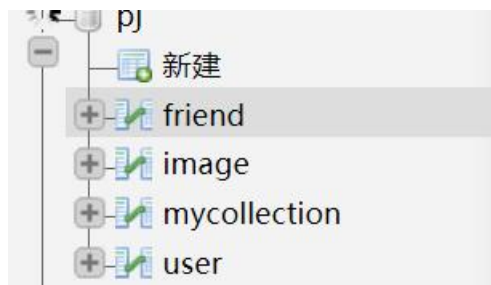
public DAO() {...}
//更新数据库
public void update(String sql,Object...args) {...}
//获取某一对象
public T get(String sql,Object...args) {...}
//获取对象列表
public List<T> getForList(String sql,Object...args) {...}
//根据某些条件获取某条性质
public <E> E getForValue(String sql,Object...args) {...}

```

而 JdbcTools 类提供了数据库的连接与释放功能。

在 ExtendTools 中，前四个类分别提供对藏品、用户信息、好友信息以及图片信息的各项操作。最后一个类是对在 ajax 从后端获取到的对象信息中的 Date 型数据格式问题。

数据库结构：



jsp 文件基本每一个对应一个页面，而 nav.jsp 作为导航文件镶嵌到每一个文件中，首先看导航，左侧是网站名，右侧未登录时展现首页、搜索、登录、注册四个按钮，登录后注册登录变为名字为用户名的下拉菜单，包括上传、我的上传、我的好友、我的收藏、登出选项。

对登录界面，包括用户名、密码、验证码的校验，前端校验符合格式后发送到 register.java

```

try {
    switch (method) {
        case "register": doRegister(customer, request, response);break;
        case "login":doLogin(customer,request,response);break;
        case "logout":doLogout(request); break;
    }
} catch (Exception e){
    e.printStackTrace();
}

```

这个文件包括处理登录注册登出的方法当登录信息转到时，会在数据库中进行比对，先判断用户是否存在，在判断密码是否准确，如不存在或不准确返回错误提示信息，并且利用 **session** 在登录界面保持原来填入的信息，对注册界面也是采用相似的方法，若注册成功进行自动登录，即调用 **doLogin** 方法。

```

HttpSession session=request.getSession();
if (customerDaoJdbc.isRegister(customer.getUserName())){
    //还要加上密码的判断
    if(customerDaoJdbc.isRegister(customer.getUserName(),customer.getPassword())) {
        System.out.println("登陆成功");
        session.setAttribute( s: "success", o: "登陆成功");
        customer.setUID(customerDaoJdbc.getUID(customer.getUserName()));
        session.setAttribute( s: "user",customer);
        //转到登陆前的界面
        response.sendRedirect(request.getSession().getAttribute( s: "path").toString());
    } else{...}
} else{...}

```

在首页除去导航外，还有轮播展示最热图片，底下还包括最新图片展示，轮播采用 **bootstrap** 插件，至于图片的获取，在最开始的时候我并不熟悉 **ajax**，所以采用向后端 **query.java** 进行请求重定向的方式获取图片

```

List<TravelImage> list=imageDaoJdbc.getHeat();
req.setAttribute( s: "img",list);
List<TravelImage> list1=imageDaoJdbc.getNewest();
req.setAttribute( s: "ns",list1);
req.getRequestDispatcher( s: "home.jsp").forward(req,resp);

```

对搜索界面，采用 **ajax** 实现分页，首先根据关键字确定符合条件的总数，然后确定总页数，每次向后端传递当前页码，获取一页的信息，返回到页面进行显示。

```
// 获取搜索关键词的数目
```

```
public long getCount(String first,String keyWord){
    String sql;
    if (first.equals("title")){
        sql="SELECT COUNT(*) FROM image WHERE title LIKE '%'+
        "'%'" ;
    }else{
        sql="SELECT COUNT(*) FROM image WHERE subject LIKE '%'+
        "'%'" ;
    }
    //获取总共的条数

    return getForValue(sql,keyWord);
}
```

```
public List<TravellImage> getSearch(int currentPage,String keyWord,String first,String second){
    int x=(currentPage-1)*6;
    String sql;
    if (first.equals("subject")&&second.equals("heat")){
        sql="SELECT imageID,imageFileName,title,subject,timeReleased,heat FROM image WHERE subject LIKE '%'+
        "'%'" ORDER BY heat DESC LIMIT ?,6";
    }else if (first.equals("title") && second.equals("heat")){
        sql="SELECT imageID,title,imageFileName,subject,timeReleased,heat FROM image WHERE title LIKE '%'+
        "'%'" ORDER BY heat DESC LIMIT ?,6";
    }else if(first.equals("subject")&&second.equals("timeReleased")) {...}else {...}

    return getForList(sql,keyWord,x);
    //获取某一页的信息
}
```

```
// 获取摘要界面图片的数目
```

在之后的我的好友、我的上传、我的收藏界面所采用的方法与之大同小异，不再赘述。

在查看好友收藏的时候，对 session 中存储登陆者信息的 UID 与进入收藏的 userID 进行对比，若不同的话对取消收藏按钮进行隐藏，如相同，可提供取消收藏服务，这并不会删除藏品，而在我的上传界面是真的会删除数据库中图片的信息。

在详情界面，若登录后会显示收藏按钮，在进入的一瞬间会通过 ajax 确定是否已经收藏，在此界面，如登录会设置 Cookie 来显示足迹，在我的收藏界面可以见到：


```

<%
    TravelImage detail=(TravelImage)request.getAttribute("detail");
    String IT=detail.getImageID()+","+detail.getTitle();
    Cookie[] cookies=request.getCookies();
    List<Cookie> bc=new ArrayList<>();
    Cookie tmp=null;

    if (cookies!=null&&cookies.length>0){
        for (Cookie cookie:cookies){
            String cn=cookie.getName();
            if (cn.startsWith("R_")){...}
        }
    }

    if (bc.size()>10&&tmp==null){
        tmp=bc.get(0);
    }

    if (tmp!=null){
        tmp.setMaxAge(0);
        response.addCookie(tmp);
    }

    Cookie cookie=new Cookie("R_"+detail.getImageID(),IT);
    response.addCookie(cookie);
%>

```

这基本复用的视频提供的方法，只不过将信息存为 imageID，title，用 split() 方法加以区分。

对上传界面，本次实际上将图片存放到了本地，仅仅将图片的各类信息上传到了数据库，这个针对是点击的上传还是修改写了两个表格

```

<c:choose>
    <c:when test="${sessionScope.user!=null&&sessionScope.imageID1==null}">
        <jsp:include page="nav.jsp"></jsp:include>
        <img id="pre" height="150px" width="150px" alt="" style="..." >
        <div class="container" style="..." >
            <form role="form" method="post" class="form-horizontal" style="..." >
                <div class="form-group" ...>
                <div class="form-group" ...>
                <div class="form-group" ...>
                <div class="form-group" ...>
                <div class="form-group" ...>
                <div class="form-group" ...>
                <div class="form-group" ...>
                <button type="submit" class="btn btn-default" style="..." onclick="upload(
                <button type="reset" class="btn btn-default" style="...">重置</button>
            </form>
        </div>
    </c:when>

```

点击修改的时候会自动填写已经保存的信息。

四、遗留问题

遇到过许多问题，比方说前后端用 `ajax` 传日期时的格式问题等，这些都通过查百度解决了，但还有一些因为时间等问题尚没有得到合理解决办法，如没有实现根据用户名搜索添加好友以及设置好友访问权限的功能，在之后的实践中应当加以完善。