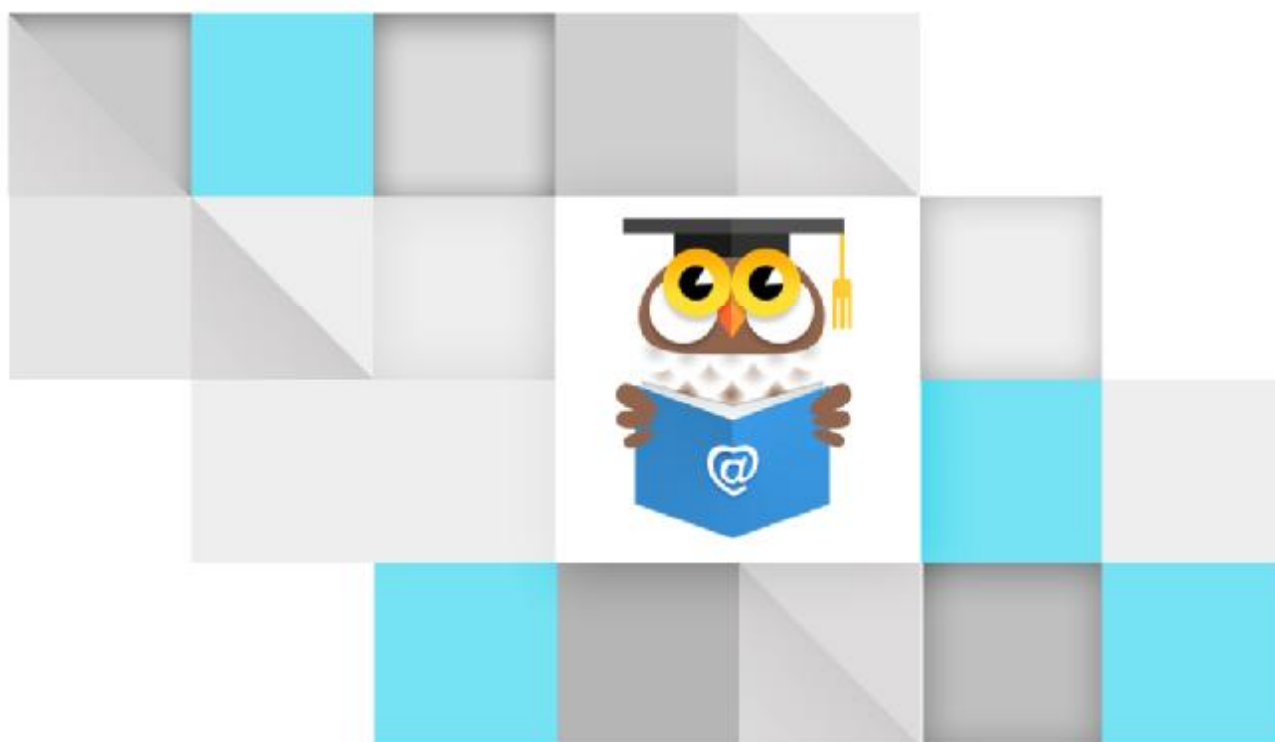


去哪玩旅游网（动态 WEB）

教学指导手册

PRJ-WTP-JEE-006 – 上传用户头像



Campus Solution Group

目 录

一、场景说明	2
1、完成效果	2
2、业务描述	2
3、实现思路	3
4、核心组件	3
二、实训技能	4
1、重点演练	4
2、相关技能	4
3、相关知识点	4
4、前置条件	4
5、搭建环境	5
三、场景任务	6
任务 1、创建并配置 Servlet 组件	6
任务 2、发送头像上传请求	8
任务 3、接收并存储上传的头像	10

一、场景说明

1、完成效果



图 1-1-1

2、业务描述

本场景主要用于实现《去哪儿玩旅游网》中，**个人信息编辑页面的头像上传功能**。

2-1. 将《去哪儿玩旅游网》的个人信息编辑页面设置为站点首页。

2-2. 通过“头像编辑”功能上传或替换个人头像：

2-2.1. 图片上传：头像上传服务器并替换：**站点根目录/user_img/1.jpg**文件。

2-2.2. 更新头像：头像上传服务器并替换原始文件后，个人信息页可显示最新头像。

注意：本场景不考虑如何实现图片预览和图片裁剪功能。

3、实现思路

3-1. 本场景建议将“头像上传功能”分为三个任务依次实现：

3-1.1. 任务1. 创建Servlet组件，该Servlet负责接收并存储客户端上传的头像。

3-1.2. 任务2. 确保“个人信息页”的头像上传请求，可发送到任务1创建的Servlet。

3-1.3. 任务3. 配置文件上传环境，实现个人头像的上传与页面动态替换。

4、核心组件

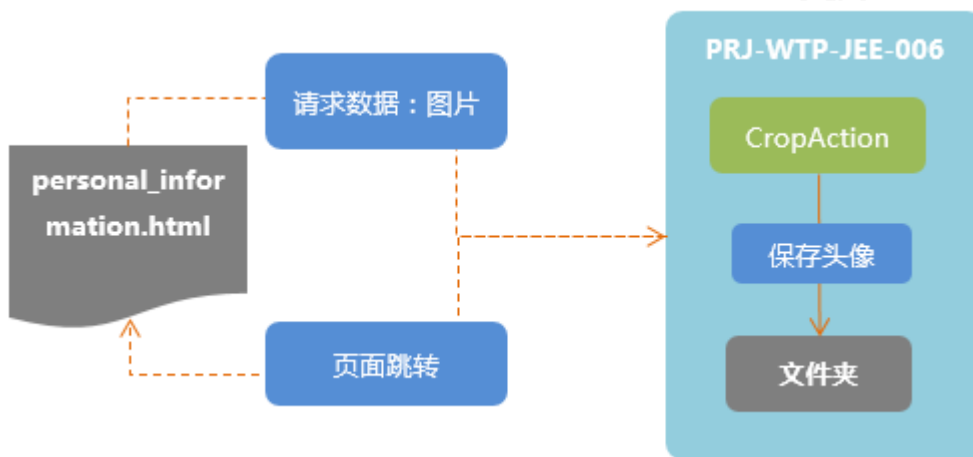


图 1-4-1

4-1. personal_information.html：个人信息编辑页面（已提供）

4-2. CropAction：处理头像上传的Servlet组件（需实现）

4-3. 文件夹（已提供）：

4-3.1. 用于存放用户头像图片的文件夹。

4-3.2. 当前场景固定存放头像的文件夹路径为：[/PRJ_WTP_JEE_006/user_img](#)。

4-3.3. 当前场景头像的固定文件名为：[1.jpg](#)

二、实训技能

1、重点演练

- 1-1. 配置文件上传的环境（按步骤实现）。
- 1-2. 体验文件上传的开发流程。
- 1-3. 通过运用请求流对象：ServletInputStream获取二进制流。
- 1-4. 通过运用Java IO技术实现数据写入。

2、相关技能

- I Servlet 技术应用

3、相关知识点

- I 开发 Servlet 组件
- I 配置 Servlet 组件
- I 接收客户端请求
- I 获取表单数据
- I 获取请求消息流

4、前置条件

- 4-1. 前置场景：PRJ-WTP-JEE-005 – 校验验证码
- 4-2. 已学技能：
 - 4-2.1. Servlet应用技术。

5、搭建环境



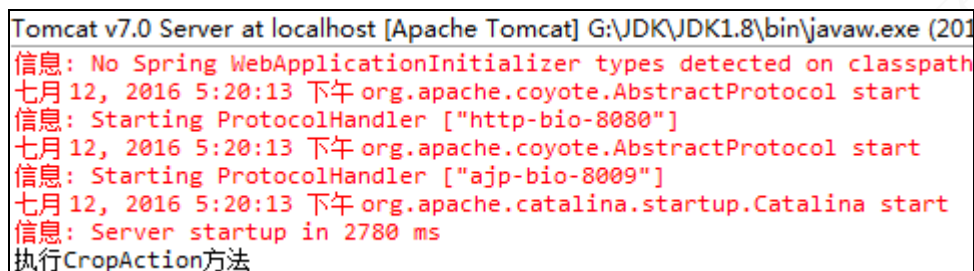
上海尚强信息科技有限公司 版权所有

三、场景任务

任务 1、创建并配置 Servlet 组件

1. 任务说明：

1-1. 完成效果：



```
Tomcat v7.0 Server at localhost [Apache Tomcat] G:\JDK\JDK1.8\bin\javaw.exe (2016-07-12 17:20:13)
信息: No Spring WebApplicationInitializer types detected on classpath
七月 12, 2016 5:20:13 下午 org.apache.coyote.AbstractProtocol start
信息: Starting ProtocolHandler ["http-bio-8080"]
七月 12, 2016 5:20:13 下午 org.apache.coyote.AbstractProtocol start
信息: Starting ProtocolHandler ["ajp-bio-8009"]
七月 12, 2016 5:20:13 下午 org.apache.catalina.startup.Catalina start
信息: Server startup in 2780 ms
执行CropAction方法
```

图 3-1-1

1-2. 任务目标：

1-2.1. 创建Servlet组件，该Servlet负责接收并存储客户端上传的头像。

1-3. 任务要求：

1-3.1. 创建用于处理头像上传的Servlet组件：

- 1) 限制1. Servlet取名：**CropAction**。
- 2) 限制2. **CropAction**创建于**campsg.qunawan.action**包中。
- 3) 限制3. 允许通过**crop.jhtml**结尾的URL访问该Servlet。

1-3.2. 当**CropAction**获取请求后，在控制台输出：“**执行CropAction方法**”。

2. 实现思路：

2-1. 在**campsg.qunawan.action**包中创建一个Class类，命名：**CropAction**。

2-2. 使**CropAction**继承HttpServlet组件。

2-3. 在**web.xml**中配置该Servlet，将url-pattern设置为：**/crop.jhtml**

2-4. 为了便于测试，将crop.jhtml设置为站点首页。

2-5. 在CropAction的doGet中，编写控制台输出测试语句。

2-6. 实现流程如下图：



图 3-1-2

3. 推荐步骤：

3-1. 创建名为CropAction的Servlet。

3-2. 配置CropAction。

3-3. 将crop.jhtml设置为首页。

4. 验证与测试：

4-1. 运行项目工程。

4-2. 在浏览器中输入 `http://localhost:8080/PRJ_WTP_JEE_006/` 并敲击回车访问。

4-3. 控制台应显示“执行CropAction方法”的字样。

任务 2、发送头像上传请求

1. 任务说明：

1-1. 完成效果：

1-1.1. 进入个人信息页：

个人资料编辑 头像编辑

* 昵称：

性别： ☐ 男 ☒ 女

* 手机号：

* 邮箱：

* 真实姓名：

* 身份证：

生日： 一年-- 一月-- 一日--

地区： ---请选择省份--- ---请选择城市---

保存

图 3-2-1

1-1.2. 进入头像编辑页面：

个人资料编辑 头像编辑

设置您的头像：

头像预览 (大小: 200 * 200)

上传照片

仅支持GIF、JPG、PNG图片

图 3-2-2

1-1.3. 请求发送到CropAction：

```
Tomcat v7.0 Server at localhost [Apache Tomcat] G:\JDK\JDK1.8\bin\javaw.exe (2016-07-12 17:20:13)
信息: No Spring WebApplicationInitializer types detected on classpath
七月 12, 2016 5:20:13 下午 org.apache.coyote.AbstractProtocol start
信息: Starting ProtocolHandler ["http-bio-8080"]
七月 12, 2016 5:20:13 下午 org.apache.coyote.AbstractProtocol start
信息: Starting ProtocolHandler ["ajp-bio-8009"]
七月 12, 2016 5:20:13 下午 org.apache.catalina.startup.Catalina start
信息: Server startup in 2780 ms
执行CropAction方法
```

1-2. 任务目标：

1-2.1. 确保“个人信息页”的头像上传请求，可发送到任务1创建的Servlet。

1-3. 任务要求：

1-3.1. 将个人信息编辑页面：personal_information.html设置为首页。

1-3.2. 通过个人信息页面的“头像编辑”功能上传“个人头像”。

1-3.3. 确保个人头像的上传请求发送到CropAction。

注意：本场景只考虑头像上传，无需考虑如何实现头像图片裁剪功能。

2. 实现思路：

2-1. 配置web.xml文件，将personal_information.html设置为首页。

2-2. 设置personal_information.html的表单（form）标签，使其满足：

2-2.1. 发送请求，将form标签的action属性设置为：/项目名/crop.jhtml。

2-2.2. 实现上传，将form标签的enctype属性设置为：multipart/form-data。

3. 推荐步骤：

3-1. 按“任务要求”与“实现思路”完成本任务。

4. 验证与测试：

- 4-1. 运行项目工程。
- 4-2. 通过http://localhost:8080/PRJ_WTP_JEE_006/访问站点，应见【完成效果图1】。
- 4-3. 点击“头像编辑”标签，应见【完成效果图2】。
- 4-4. 点选“上传照片”按钮，选择任意一张图片在页面中显示图片预览。
- 4-5. 点击“保存”按钮，请求将会发送到CropAction，应见【完成效果图3】。

任务 3、接收并存储上传的头像

1. 任务说明：

1-1. 完成效果：

图 3-3-1

1-2. 任务目标：

- 1-2.1. 配置文件上传环境，实现个人头像的上传与页面动态替换。

1-3. 任务要求：

- 1-3.1. 确保客户端用户选择的个人头像图片数据能够被CropAction接收。

- 1) 限制1. 头像上传必须通过请求流实现，不得使用第三方框架技术。

1-3.2. CropAction负责将头像存储在服务器端。

1) 限制1. 上传的头像图片固定替换：**站点根目录/user_img/1.jpg**

1-3.3. 确保个人信息页面能够看到上传后的新头像。

2. 实现思路：

2-1. 配置二进制文件（图片）的上传环境（按步骤实现）。

2-2. 获取Servlet的请求流对象：ServletInputStream。

2-3. 利用IO中的FileOutputStream流对象存储上传的个人头像。

2-3.1. 图片文件固定替换：站点根目录/user_img/1.jpg

2-4. 由于个人信息页面的头像显示控件始终读取1.jpg，因此上传的头像替换1.jpg后，即可实现个人信息页头像的动态替换功能。

2-5. 头像上传完毕后，跳转到personal_information.html页面。

2-6. 实现流程如下图所示：

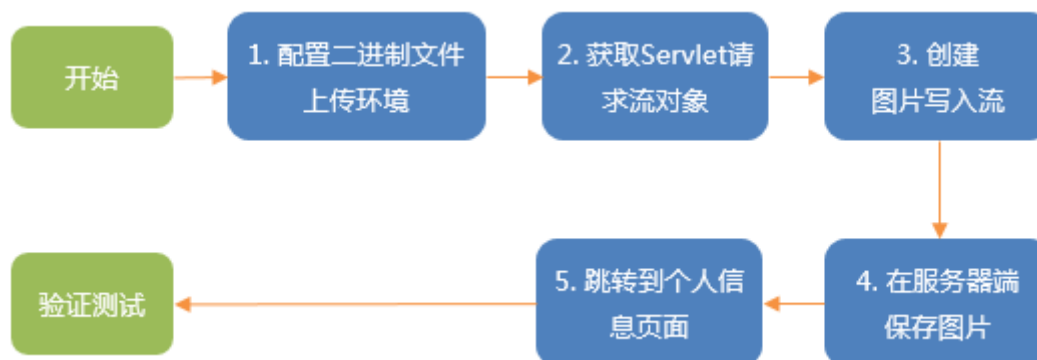


图 3-3-2

3. 推荐步骤：

3-1. 配置二进制文件（图片）的上传环境（按步骤实现）：

3-1.1. 为Tomcat的context.xml文件增加上传属性。

1) 打开Eclipse中Server工程文件夹。

2) 打开Tomcat v7.0 Server at localhost-config/context.xml。

3) 找到<Context>节点，添加allowCasualMultipartParsing属性，值为true。

+ 提示：

1) allowCasualMultipartParsing属性说明：

1-1. 该属性的缺省值为false；

1-2. 当设置为true时，Servlet可以通过HttpServletRequest的getPart方法获取表单中的二进制数据（文件、图片等）。

3-2. 获取Servlet请求流对象

3-2.1. 在CropAction的doGet方法中获取请求对象。

+ 提示：

1) 获取请求流对象的方法如下：

```
/*
 * getPart方法可以方便的获取表单中的上传数据内容。
 * getPart方法的参数为：个人信息页面上传控件的name属性。
 * getInputStream用于返回请求流对象：ServletInputStream。
 * 请求流对象ServletInputStream继承自InputStream
 */
InputStream is = request.getPart("image").getInputStream();
```

3-3. 创建图片写入流：

3-3.1. 获取上传头像的存储路径：

+ 提示：

1) 获取上传头像的存储路径：

```
/*
 * getServletContext(): 获取当前站点对象。
 * getRealPath(): 获取当前站点的根目录。
 * getRealPath的参数：将站点根目录与参数路径合并组成完整的路径并返回。
 * 返回结果：http://localhost:8080/PRJ_WTP_JEE_006/user_image/1.jpg
 */
String filePath =
```

```
request.getServletContext().getRealPath("/user_image/1.jpg");
```

3-3.2. 利用FileOutputStream写入头像图片数据。

3-3.3. 将从请求流对象中获取的头像数据写入：`/user_image/1.jpg`文件中。

3-3.4. 关闭文件写入流FileOutputStream。

+ 提示：

1) 读取头像图片数据，并写入`/user_image/1.jpg`文件中的思路：

1.1 循环读取请求流对象中的数据，每次读取1024字节长度。

1.2 返回结果不等于-1继续读取，否则退出。

1.3 利用FileInputStream将读取的数据写入：`/user_image/1.jpg`文件中。

```
while((i = is.read(buf)) != -1){  
    fos.write(buf, 0, i);  
}
```

3-4. 数据写入结束后，跳转到个人信息页面：`personal_information.html`

4. 验证与测试：

4-1. 运行项目工程。

4-2. 在浏览器地址栏中输出 `http://localhost/PRJ_WTP_JEE_006/` 进行访问。

4-3. 在个人信息页右侧栏顶部找到“头像编辑”并点击，进入头像编辑界面。

4-4. 点击“上传照片”，选择一个图片文件后确定。

4-5. 点击“保存”按钮后，左侧栏的用户头像发生变化。

注意：由于浏览器缓存原因，如果头像未更新，点击浏览器的刷新按钮刷新当前页面即可。

作者：Roger.Huang