

# 仿 airdrop 功能实现

——成员：谭慧、胡珂

## 一. 关键功能

### 1.1 文件传输

在桌面端编写了一个 index(jsp)网页，利用 Tomcat 运行充当服务器，当手机与电脑处在同一个局域网时，可以通过访问网页的方式向电脑发送文件，保存在电脑设置好的目录中。

### 1.2 文本共享

在文件传输的基础上，于 jsp 网页上添加了复制粘贴模块，可以在网页上显示共享剪切板内容。手机访问网页后，会监听手机的复制操作，手机复制后，会自动获取设备的剪切板内容，电脑可以直接右键或者利用快捷键 Ctrl-V 进行粘贴。网页上显示的共享剪切板内容也会及时进行更新。电脑进行复制操作后，会自动更新共享剪切板，手机也可实现一键复制共享剪切板中的内容。

## 二. 关键代码

### 2.1 文件传输

#### 2.1.1 确定文件保存文件夹

```
// 创建上传文件的保存路径, 建议在WEB-INF路径下, 安全, 用户无法直接访问上传的文件;  
String uploadPath =this.getServletContext().getRealPath( s: "/WEB-INF/upload");  
File uploadFile = new File(uploadPath);  
if (!uploadFile.exists()){  
    uploadFile.mkdir(); // 创建这个目录  
}
```

图 1 创建文件夹

如图 1, 我们通过 `this.getServletContext().getRealPath(“路径”)` 方法获得创建上传文件的保存路径。如果目录不存在那么创建此目录。在选择目录路径时, 这里我们选择了 `WEB-INF` 目录来保存上传文件, 因为在此路径下, 较为安全, 用户无法直接访问上传的文件。此外, 我们还创建了临时路径, 假如文件超过了预期的大小, 我们就把他放在一个临时文件中, 过几天自动删除, 或者提醒用户转为永久。

#### 2.1.2 处理上传文件

```
public static String uploadParseRequest(ServletFileUpload upload, HttpServletRequest request, String uploadPath)  
    throws FileUploadException, IOException {  
    String msg = "";  
    // 把前端请求解析, 封装成FileItem对象  
    List<FileItem> fileItems = upload.parseRequest(request);  
    for (FileItem fileItem : fileItems) {  
        //fileItem 每一个表单对象  
        if (fileItem.isFormField()) { // 判断上传的文件是普通的表单还是带文件的表单  
            // getFieldname指的是前端表单控件的name;  
            String name = fileItem.getFieldName();  
            String value = fileItem.getString( s: "UTF-8"); // 处理乱码  
            System.out.println(name + ": " + value);  
        } else { // 判断它是上传的文件  
            // ===== 处理文件=====  
            // 拿到文件名  
            String uploadFileName = fileItem.getName();  
            System.out.println("上传的文件名: " + uploadFileName);  
            if (uploadFileName.trim().equals("") || uploadFileName == null) {  
                continue;  
            }  
        }  
    }  
}
```

图 2 处理上传文件部分代码

如图 2，我们在前端进行文件上传时，后端需要把前端的请求解析，封装成 `FileItem` 对象，并判断上传的文件是普通的表单还是带文件的表单。如果是带文件的表单，判断上传了文件，那么首先获取文件名及文件后缀名，如果文件后缀名不是我们所需要的，就直接 `return`。不进行处理,告诉用户文件类型不对。获得文件上传的流，并判断是否上传完毕，待上传成功后，关闭文件上传流。这些操作如果直接使用 `Servlet` 获取上传文件的输入流然后再解析里面的请求参数较为麻烦，因此这里我们依赖了 `Apache` 的 `common-fileupload` 文件上传组件来实现。

## 2.2 复制粘贴

```
document.addEventListener('copy',function(e){
    // 获得clipboardData 对象
    let clipboardData = e.clipboardData || window.clipboardData;
    var form=document.getElementById("clipboard_form");
    // 如果 未复制或者未剪切，直接 return
    if(!clipboardData) return ;
    // 获得 Selection 对象 表示用户选择的文本范围或光标的当前位置。
    // 将用户输入的剪切或者复制的文本转化为字符串
    let text = window.getSelection().toString();
    if(text){
        // 如果文本存在，首先取消默认行为
        e.preventDefault();
        console.log("可以获取",text);
        document.getElementById("clipboard").value=text;
        form.submit();
        // alert('复制成功');
    }
})
```

图 3 处理共享剪贴板部分代码

我们设置了监听器，对复制事件进行监听，当手机或者电脑出现了复制事件后，通过 `js` 自动获取当前剪贴板的内容，向服务器发送请求，通过 `application` 对象设置共享剪贴板 `clipboard`，并将当前系统剪贴板中的内容保存到共享剪贴板

clipboard 中，实现所有访问客户的共享。

```
function copyToClipboard(text, callback) {  
    // 实现复制方法一：  
    if (navigator.clipboard) {  
        navigator.clipboard.writeText(text);  
        callback && callback(true);  
        return;  
    }  
    // 实现复制方法二：  
    var copyInput = document.getElementById('COPY_INPUT');  
    if (!copyInput) {  
        copyInput = document.createElement('input');  
        copyInput.setAttribute('id', 'COPY_INPUT');  
        copyInput.style.position = 'fixed';  
        copyInput.style.left = '-100%';  
        copyInput.style.top = '0';  
        copyInput.style.zIndex = -100;  
        copyInput.style.opacity = 0;  
        document.body.appendChild(copyInput);  
    }  
    copyInput.value = text;  
    copyInput.focus();  
    copyInput.select();  
    // document.execCommand 可能会被废弃  
    if (document.execCommand('copy')) {  
        document.execCommand('copy');  
    }  
    copyInput.blur();  
    callback && callback(true);  
}
```

图 4 处理共享剪贴板部分代码

当用户进入页面后，通过 js 实现自动将共享剪贴板中的内容复制到当前用户的剪贴板，由于不同浏览器可能会出现兼容问题，我们集合了四种复制的方式，分别为图中的 3 种方式，以及 clipboard.js。当用户进入页面时，会获取当前共享

剪贴板 clipboard 的内容，如果共享剪贴板不为空，会通过 js 的方法 copyToClipboard 将共享剪贴板 clipboard 中的内容复制到系统剪贴板，若少数浏览器不兼容，则可通过 clipboard.js 点击复制，实现一键复制。若复制成功，则会出现弹窗，显示复制成功；若复制失败，会提示用户复制失败，请手动复制。

## 三. 单元测试

单元测试是对软件基本组成单元（函数、方法）进行的测试，其目的在于发现各模块内部可能存在的各种错误，主要是针对系统最基本的单元代码进行测试。单元测试可以帮助我们快速发现潜在问题，为重构软件做准备，以最小的粒度保证代码的可靠性。

在这里我们对 getDiskFileItemFactory 方法与 getServletFileUpload 方法进行了单元测试。在对 getDiskFileItemFactory 方法进行的单元测试里创建 DiskFileItemFactory 对象，处理文件路径或大小限制，输出创建工厂。在对 getServletFileUpload 进行的单元测试里，输出了获取的 ServletFileUpload 的参数变量。

```
void getDiskFileItemFactory() {
    DiskFileItemFactory factory = new DiskFileItemFactory();
    System.out.println(factory);
}

@Test
void getServletFileUpload() {
    DiskFileItemFactory factory = new DiskFileItemFactory();
    ServletFileUpload upload = new ServletFileUpload(factory);
    System.out.println(upload);
}
```

图 5 单元测试部分代码

## 四. 使用教程

### 4.1 文件传输



图 6 上传文件

用户进入界面后点击“开始上传”，选择需要上传的文件后，文本框会显示上传的文件（考虑到安全问题，我们对文件路径实施了处理，避免隐私被泄露的风险，会生成一个假路径“C:\fakepath\”，并显示结果“C:\fakepath\”+文件名），如果上传文件正确，则点击提交，实现上传，并跳转结果页面显示上传结果。如果文件选择错误，则可以点击重置，重新选择文件。

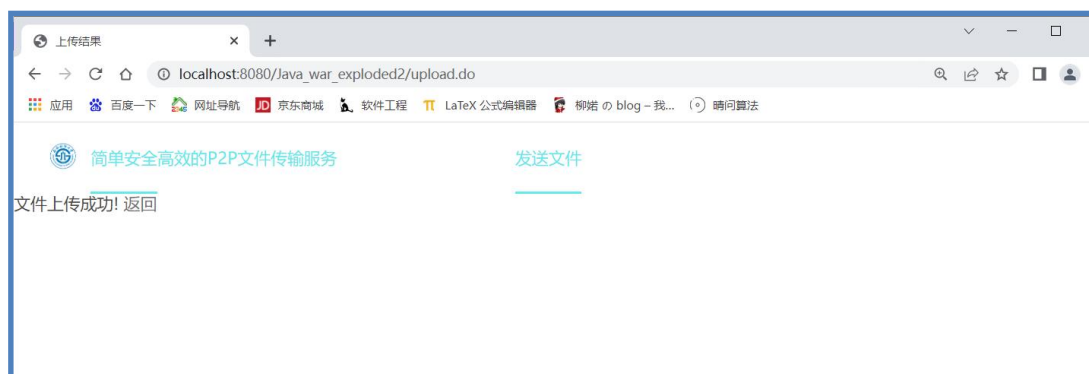


图 7 文件上传成功页面

## 4.2 复制粘贴

用户可通过 **ctrl+c** 或者右键复制，复制后页面会自动向服务器提交当前复制内容，并通过弹窗显示复制结果，同时会重新更新页面，更新共享粘贴板的值。

用户进入页面，会自动获取共享粘贴板的内容，进入页面后，若浏览器兼容，网页可直接将共享粘贴板的内容写入用户粘贴板，用户可直接通过右键或 **ctrl+v** 实现粘贴。若浏览器不兼容，则可通过点击复制按钮，实现一键复制，复制结果会用弹窗提示用户，若复制失败，同时会提示用户手动复制。

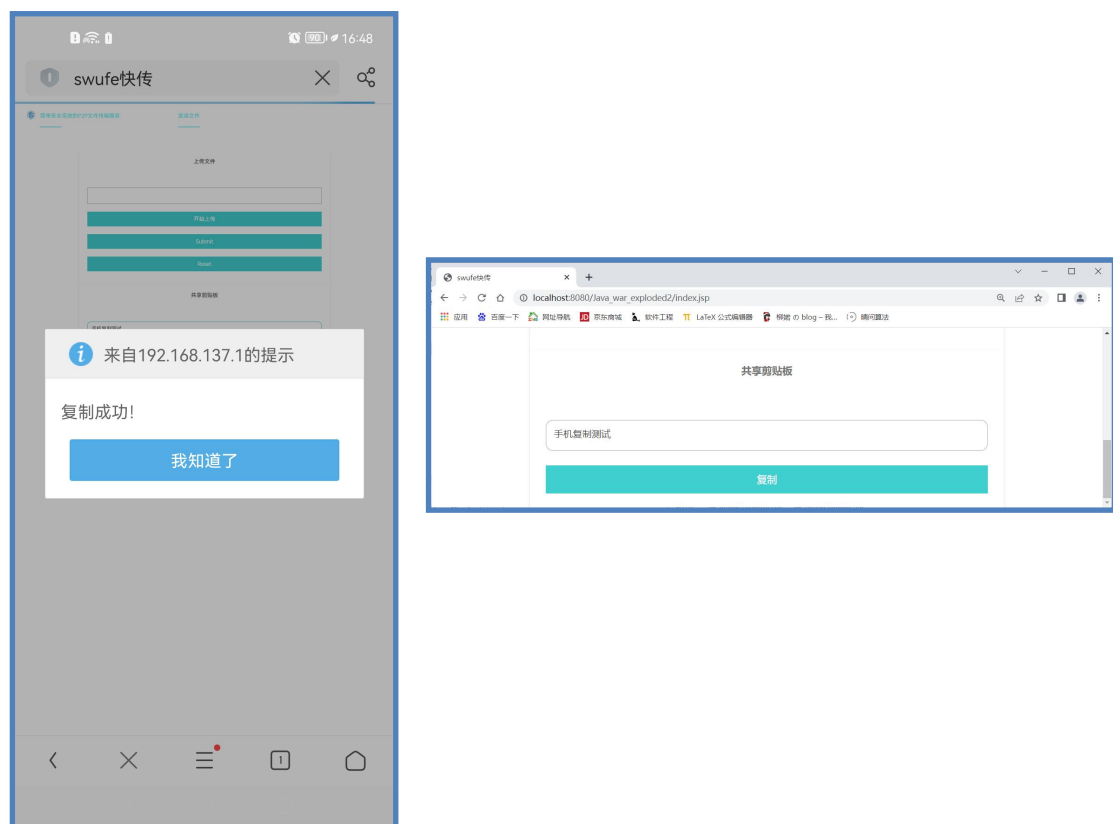


图 8 手机端复制测试

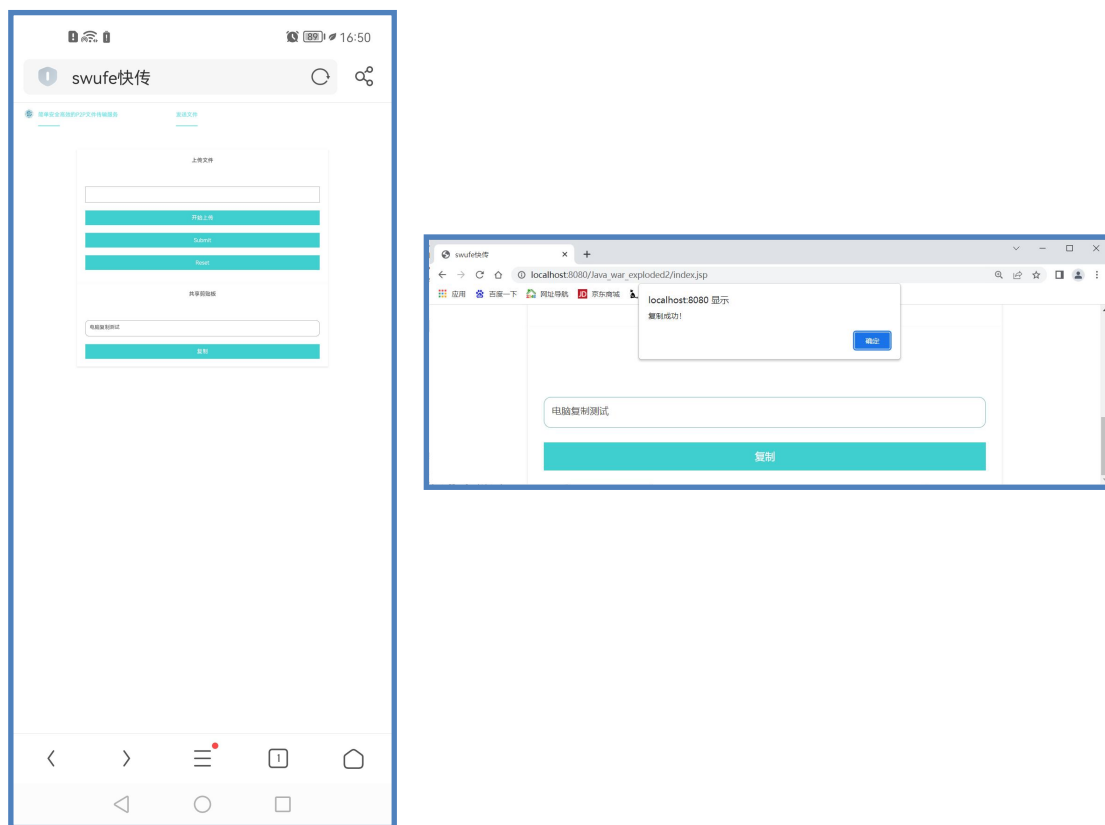


图 9 电脑端复制测试

## 五. 持续集成

持续集成是一种软件开发实践，即团队开发成员经常集成他们的工作，通常每个成员每天至少集成一次，也就意味着每天可能会发生多次集成。每次集成都通过自动化的构建（包括编译，发布，自动化测试）来验证，从而尽早地发现集成错误。同样，我们将代码定期变更合并到 GitHub 仓库中，进行 Action 操作，自动运行构建和测试操作，以便我们及时发现错误。

## 六. 亮点

页面友好：页面简洁、美观，功能能够得到完美的展示，用户使用体验良好。  
操作简单方便：无需安装客户端，只需要连接局域网，选择好想要发送的文



件，即可开始上传。

数据安全：数据自带加密，且页面隐藏真实文件路径，免去隐私泄漏的风险。

兼容不同浏览器：共享剪切板集合多种方案，在不同浏览器都可完美实现，让用户可以轻松复制，方便快捷。

## 七. 总结与反思

在进行仿 `airdrop` 功能，实现复制粘贴时，我们首先考虑直接使用 `Servlet` 获取上传文件的输入流然后再解析里面的请求参数进行处理，可是此方法较为麻烦，期间也出现了很多错误。在进行网上查阅资料后，于是我们选择了 `Apache` 的开源 `common-fileupload` 文件上传组件来实现。可惜的是，由于时间关系，我们文件传输只能由手机传送文件给电脑，无法实现文件互传。

于复制粘贴功能而言，我们在广泛查询资料后，最终选择了 `js` 的 `clipboard` 进行处理。但是发现在访问页面时，有部分浏览器不兼容，或是有多余的保护机制，不支持设备进行复制。因此我们又集合了三种不同的复制方式，并在进行直接快捷键粘贴之外，也可允许用户手动复制粘贴。

虽然目前只实现了网页界面，并未开发真正的软件。但后续，我们会努力完善这个项目，希望它能成为一个真正有意义，有价值的软件。