



# 软件工程课程论文

学年学期： 2022 年春季学期

课程名称： 软件工程

论文题目： 仿 Airdrop 软件设计

学生学号： 41911105

学生姓名： 黄婷婷

学院： 经济信息工程学院

年级专业： 计算机科学与技术

评语：

得 分：

评阅教师签字： 年 月

# 基于 Java Web 的仿 AirDrop 实现

## Java Web-based imitation AirDrop implementation

**摘要：**Java Web 是用 Java 技术来解决相关 Web 互联网领域的技术栈。Web 包括：Web 服务端和 Web 客户端两部分。Java 在客户端的应用有 Java Applet，不过使用得很少，Java 在服务器端的应用非常的丰富，比如 Servlet, JSP、第三方框架等等。Java 技术对 Web 领域的发展注入了强大的动力。苹果生态的 AirDrop 在传输文件时十分方便，本项目计划模仿实现该功能，主要利用 Servlet、JSP、开源组件 Commons-FileUpload 等技术完成仿 AirDrop 的实现，在初步实现第一阶段的客户需求的基础上，结合 clipboard.js 完成对第二阶段需求的构思和初步实践。

**关键词：**Java Web；文件传输；Clipboard

### 1 项目概述

#### 1.1 项目需求

第一阶段开发需求：在桌面端编写一个软件或者网页充当服务器，当手机和服务器处于同一个局域网时，通过网页的方式发送文件。

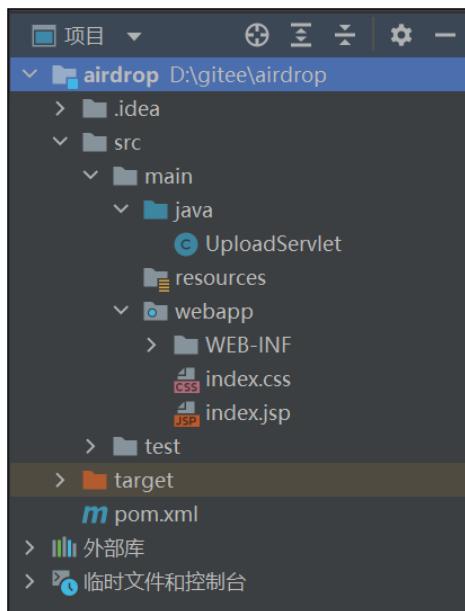
第二阶段开发需求：在第一阶段的基础上，进一步实现手机端复制文字，能够在电脑端粘贴。

#### 1.2 开发环境

IntelliJ IDEA 2021.3.2

Tomcat 8.5.53

#### 1.3 项目目录结构



项目目录结构如上图所示：

src/main/java 文件夹：存放实现业务逻辑的 java 代码（UploadServlet）

src/main/webapp 文件夹： index.css 样式表;index.jsp:用户界面

src/test 文件夹：存放测试代码

target 文件夹：存放项目构建后的文件和目录、jar 包、war 包、编译的 class 文件

## 2 关键功能及实现

本项目共两个关键功能：局域网内上传文件和共享剪贴板。

### 2.1 文件发送与接收功能

要实现 Web 开发的上传功能，通常需要完成两步操作：一是在 Web 页面中添加上传输入项；二是在 Servlet 中读取上传文件的数据，并保存到本地硬盘中。

#### 2.1.1 导入外部 jar 包

创建一个 Web 项目，导入开源组件 Commons-FileUpload 的两个 jar 包：

commons-fileupload-1.3.1.jar

commons-io-2.6.jar

由于在 Servlet 直接读取上传数据，并且解析出相应文件数据是一项非常麻烦的工作。为方便处理上传数据，利用 Apache 组织提供的开源组件 Commons-FileUpload 可以将“multipart/form-data”类型请求的各种表单域解析出来，并实现一个或多个文件上传，同时也可限制上传文件的大小等内容。

#### 2.1.2 创建上传页面 index.jsp

上传大多数情况是通过表单的形式提交给服务器，使用 <input type="file"> 标签。

使用 <input type="file"> 标签需要注意以下三点：

(1) 必须设置 name 属性，不然浏览器不会发送上传文件的数据。

<input type="text" name="name"/>

(2) 必须将 method 属性设置为 post。

(3) 普通表单提交默认 enctype="application/x-www-form-urlencoded"，但是当表单中存在文件类型时，需要设置 enctype="multipart/form-data"，它不对字符进行编码，用于发送二进制的文件（即所有文件类型，如视频、图片、音乐、文档都可以用此类型）。

<form action="UploadServlet" method="post" enctype="multipart/form-data">

上传页面如下图所示：

The screenshot shows a simple file upload interface. At the top left, it says "Uploader:". To its right is a rectangular input field for selecting files. Below this, the text "Uploading a file:" is followed by two buttons: "选择文件" (Select File) and "未选择文件" (No File Selected). At the bottom left of the form is a large, prominent "UPLOAD" button.

代码如下图所示：

```
○ ○ ○

<form action="UploadServlet" method="post" enctype="multipart/form-data">
    <table>
        <tr>
            <td>Uploader:</td>
            <td><input type="text" name="name" /></td>
        </tr>
        <tr>
            <td>Uploading a file:</td>
            <td><input type="file" name="myfile" /></td>
        </tr>
        <tr>
            <!-- 设置单元格可横跨的列数。 -->
            <td colspan="2"><input type="submit" value="UPLOAD" /></td>
        </tr>
    </table>
</form|
```

## 2.1.3 创建 Servlet

FileUpload 是 Apache 的 commons 组件提供的上传组件，它最主要的工作就是帮助解析 request.getInputStream()。FileUpload 的核心类有 DiskFileItemFactory、ServletFileUpload、FileItem。

使用 FileUpload 固定步骤：

- (1) 创建工厂类：DiskFileItemFactory factory=new DiskFileItemFactory();
- (2) 创建解析器：ServletFileUpload upload=new ServletFileUpload(factory);
- (3) 使用解析器解析 request 对象：List<FileItem> list=upload.parseRequest(request);

关键代码：

- (1) 创建 DiskFileItemFactory 工厂对象，并设置文件缓存目录为"D:\\TempFolder"，如果该文件夹不存在则新建一个，代码如下图。

```
○ ○ ○

DiskFileItemFactory factory = new DiskFileItemFactory();
File f = new File("D:\\TempFolder");
if (!f.exists()) {
    f.mkdirs();
}
factory.setRepository(f);
```

- (2) 创建 ServletFileUpload 对象，并设置字符编码，代码如下图。

○ ○ ○

```
ServletFileUpload fileUpload = new ServletFileUpload(factory);
fileUpload.setHeaderEncoding("utf-8");
```

(3) 使用解析器解析 request 对象，将 form 表单的各个字段封装为 FileItem 对象，一个 FileItem 对象对应一个表单项，代码如下图。

○ ○ ○

```
List<FileItem> fileItems = fileUpload.parseRequest(request);
PrintWriter writer = response.getWriter();
```

(4) 获取字符流，遍历 FileItem 的 List 集合，处理输出上传者 Uploader，代码如下图。

boolean isFormField(): 判断是否是普通表单字段，若是，返回 true，否则返回 false。

String getFieldName(): 获取表单项的 name 的属性值。

String getString(): 获取字段的内容。如果是普通字段，则是它的 value 值；如果是文件字段，则是文件内容。

○ ○ ○

```
List<FileItem> fileItems = fileUpload.parseRequest(request);
PrintWriter writer = response.getWriter();
for (FileItem fileItem : fileItems) {
    if (fileItem.isFormField()) {
        String name = fileItem.getFieldName();
        if (name.equals("name")) {
            if (!fileItem.getString().equals("")) {
                String value = fileItem.getString("utf-8");
                writer.print("Uploader:" + value + "<br />");
            }
        }
    } else{...}
}
```

(5) 处理输出文件名 Filename: 处理上传的文件，获取上传的文件名，并保持文件名唯一，代码如下图。

String getName(): 获取文件字段的文件名。如果是普通字段，则返回 null。

○ ○ ○

```
else {
    String filename = fileItem.getName();
    if (filename != null && filename != "") {
        writer.print("Filename:" + filename + "<br />");
        filename = UUID.randomUUID().toString() + "_" + filename;
    }
}
```

(6) 创建文件路径，将上传文件保存在/upload/文件夹中。获得文件上传流，使用 FileOutputStream 打开服务器端的上传文件，读取上传文件的字节，并将其输出到服务器端的上传文件输出流中。代码如下图。

InputStream getInputStream(): 获得文件内容的输入流。如果是普通字段，则返回 value 值的输入流。

```
○○○  
String webpath = "/upload/";  
String filepath = getServletContext().getRealPath(webpath + filename);  
File file = new File(filepath);  
file.getParentFile().mkdirs();  
file.createNewFile();
```

```
○○○  
InputStream in = fileItem.getInputStream();  
FileOutputStream out = new FileOutputStream(file);  
byte[] bytes = new byte[1024];  
int len;  
while ((len = in.read(bytes)) > 0)  
    out.write(bytes, 0, len);  
in.close();  
out.close();  
fileItem.delete();  
writer.print("The file upload was successful!");
```

## 2.2 共享剪贴板功能

Clipboard 接口实现了 Clipboard API，如果用户授予了相应的权限，其就能提供系统剪贴板的读写访问能力。在 Web 应用程序中，Clipboard API 可用于实现剪切、复制和粘贴功能。

主要步骤如下：

(1) 引入一个第三方 CDN 获取 clipboard.min.js。

```
○○○  
<script src="https://cdnjs.cloudflare.com/ajax/libs/clipboard.js/2.0.10/clipboard.min.js">  
</script>
```

(2) 设置三个 button: Copy\_Textarea, Copy2, Copy3。

通过给目标元素添加一个 data-clipboard-target 属性，这三个 button 可以从另一个元素复制内容。

Copy\_Textarea 复制的是 textarea 里的内容；

Copy2 复制的是 div2 里的内容；

Copy3 复制的是 div3 里的内容。

```
○ ○ ○

<textarea rows="10" cols="30" id="bar"></textarea>
<button class="btn" data-clipboard-action="copy" data-clipboard-target="#bar">Copy_Textarea</button>
<br>
<div2>copy-----2</div2>
<button class="btn" data-clipboard-action="copy" data-clipboard-target="div2">Copy2</button>
<br>
<div3>copy-----3</div3>
<button class="btn" data-clipboard-action="copy" data-clipboard-target="div3">Copy3</button>
```

(3) 实例化构造函数。

```
○ ○ ○
```

```
var btns = document.querySelectorAll('button');
var clipboard = new ClipboardJS(btns);
```

(4) 操作成功或者失败时的行为：

```
○ ○ ○
```

```
clipboard.on('success', function (e) {
    console.log(e);
});
clipboard.on('error', function (e) {
    console.log(e);
});
```

以上操作能实现在电脑端将有限的预置的网页文本复制到剪贴板的功能，但这只是共享粘贴板的初步探索实践，要实现在不同设备（Android 与 Windows）间共享剪贴板还有很多工作，由于个人能力限制，尚未能完整实现功能。

经搜索资料，网络上有一款的 Windows 平台应用实现了这个功能，clipboard-online 将本机剪贴板暴露于 HTTP 端口，其他设备通过发送 HTTP 请求来获取、发送剪贴板。而其他设备发送剪贴板后，PC 可以直接粘贴。现已支持文本、文件、图像。后面的开发阶段将学习该开源优秀项目的实现来进一步完善本项目的“共享粘贴板”功能的开发。



注册 登录

Clipboard-online：在不同设备之间分享剪贴板。附 iOS 和 Android 设备分享方法。

讨论分享 android ios windows

### 3 单元测试

由于 web 项目的特殊性，一般来说，单元测试会集中在业务逻辑层，需要做单元测试的地方是：逻辑复杂的功能模块。单元测试的模块必须是个闭合的系统，有固定的输入和输出。因此在系统设计阶段就应该进行充分的考虑：代码的可测试性。要做到代码的可测试性主要有以下几个要点：

- (1) 把复杂的问题抽象为模型的能力。
- (2) 好的程序架构，即程序要模块化。单元测试多是针对类或者函数进行。单元测试要求测试对象是个闭合的系统。
- (3) 因为 web 程序的特殊性，有时候，要做到完全闭合会很困难。那这时候，适当的用一些小技巧来实现可测试是必要的。测试的目的是为了保证产品质量，不能因为单元测试延误工期，本末倒置。具体实现上可以定义一个环境常量，当这个环境常量等于测试模式的时候，就可以做一些特殊的处理。

Junit 是一个 Java 语言的回归测试框架，Junit 测试也是程序员测试，即所谓的白盒测试，它需要程序员知道被测试的代码如何完成功能，以及完成什么样的功能。

对于要在项目里对 servlet 进行单元测试，可以使用 junit 进行测试。

尝试编写 UploadServlet 时，使用 Junit 测试，可用的方法有：

- (1) 使用 Cactus 框架

Cactus 是一个简单的测试框架，用于对服务器端 Java 代码（Servlet、EJB、标签库、过滤器等）进行单元测试。Cactus 的目的是降低为服务器端代码编写测试的成本。它使用 JUnit 并对其进行扩展。Cactus 实现了容器内策略，这意味着测试在容器内执行。

- (2) 创建一个嵌入式服务器来“托管”servlet，允许您使用旨在调用实际服务器的库来编写针对它的调用（这种方法的有用性在某种程度上取决于您可以轻松地将“合法”程序化对服务器的调用。

- (3) 使用 httpunit + servlet 测试

- (4) 使用 easymock + servlet

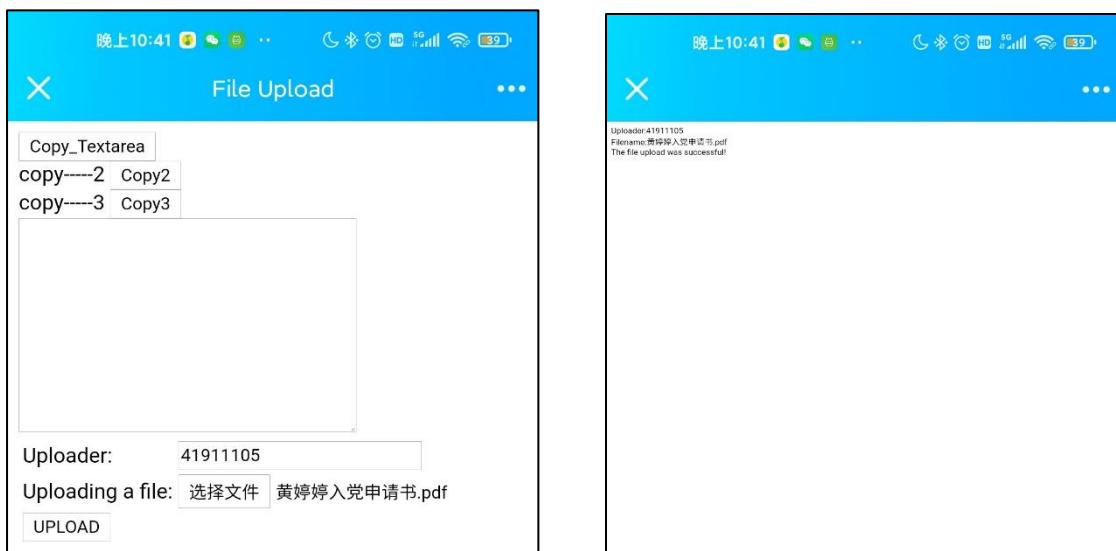
因为以上方法参考学习代码均使用 request.getParameter()，通过预置参数的方法对 doGet/doPost 测试，但本项目接收时不能再用 request.getParameter()，而是使用 request.getInputStream() 解析二进制流，得到 ServletInputStream 对象，以上几种尝试编写单元测试在实践均遇到了困难。

由于 web 程序和一般的软件开发不同，自动化测试的效率和必要性一直较低，因此人工测试一直是 web 项目的最主要测试手段。对于 web 项目而言，自动化测试可以分为单元测试和功能测试。功能测试主要针对具体页面进行测试，既然是针对具体页面进行测试，结合本项目的特性，采用人工测试的方式也很直接，高效，且灵活。故本

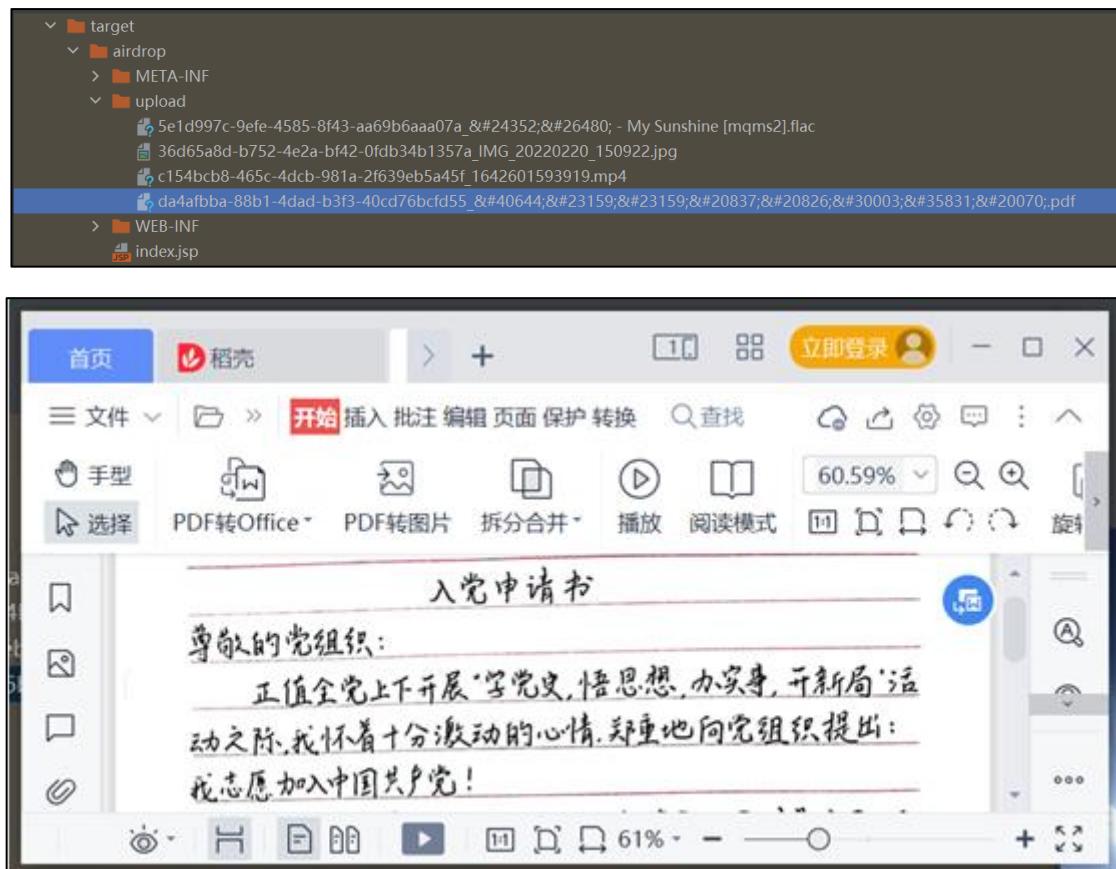
项目的测试阶段主要采用人工测试的方式，测试视频、图片、音乐、文档四种主流文件格式。

### 3.1 文档测试

#### (1) 上传文档

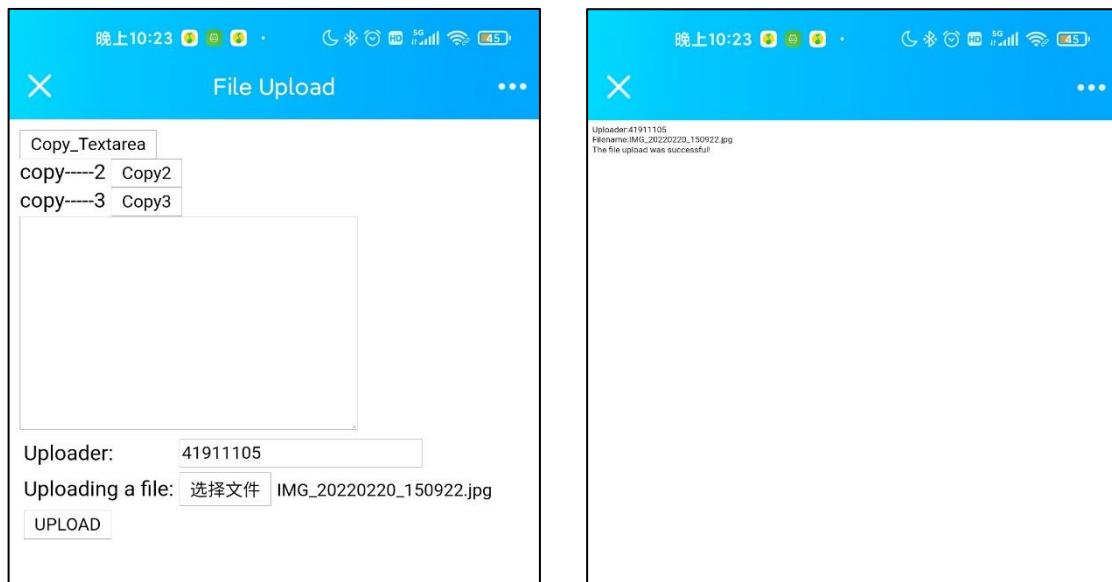


#### (2) 双击打开文档

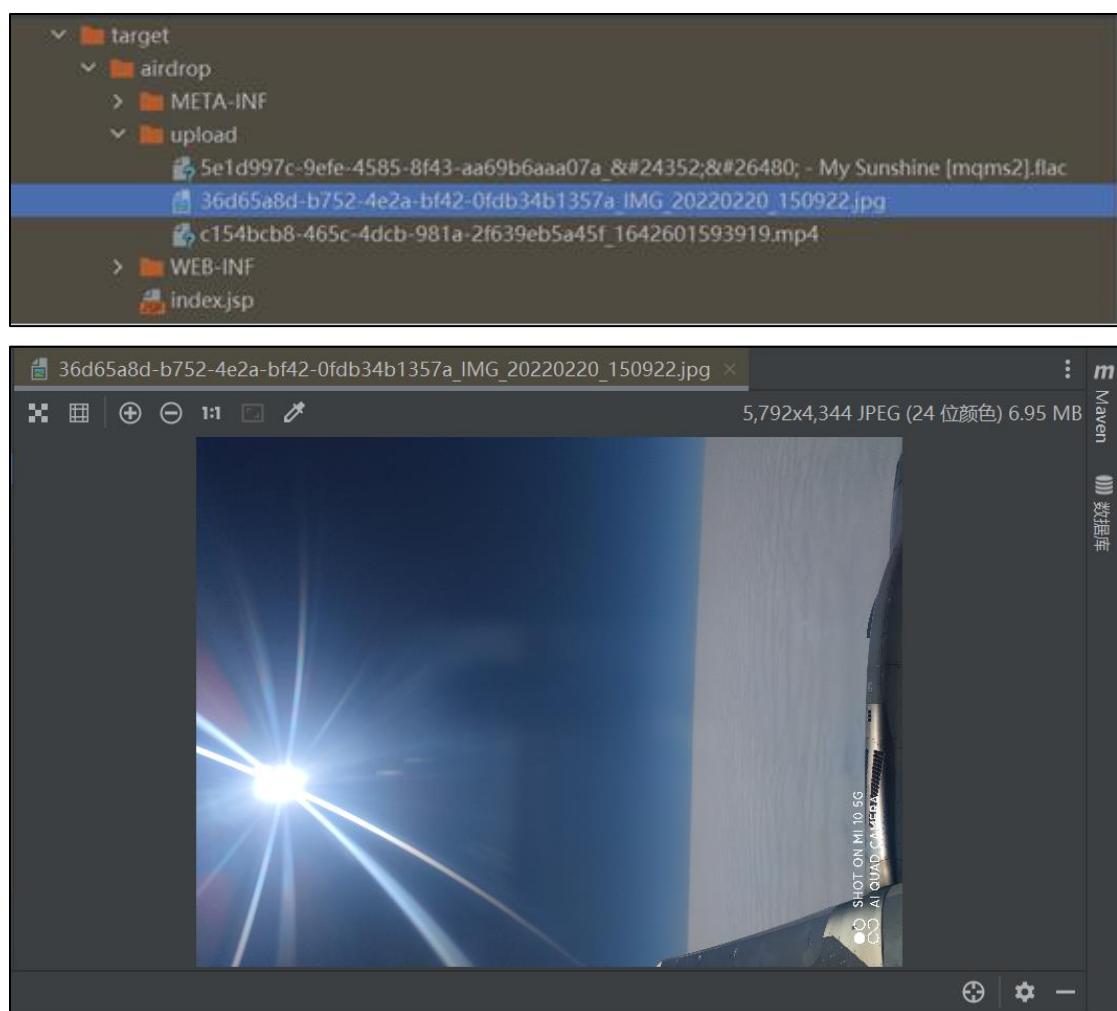


### 3.2 图片测试

#### (1) 上传图片

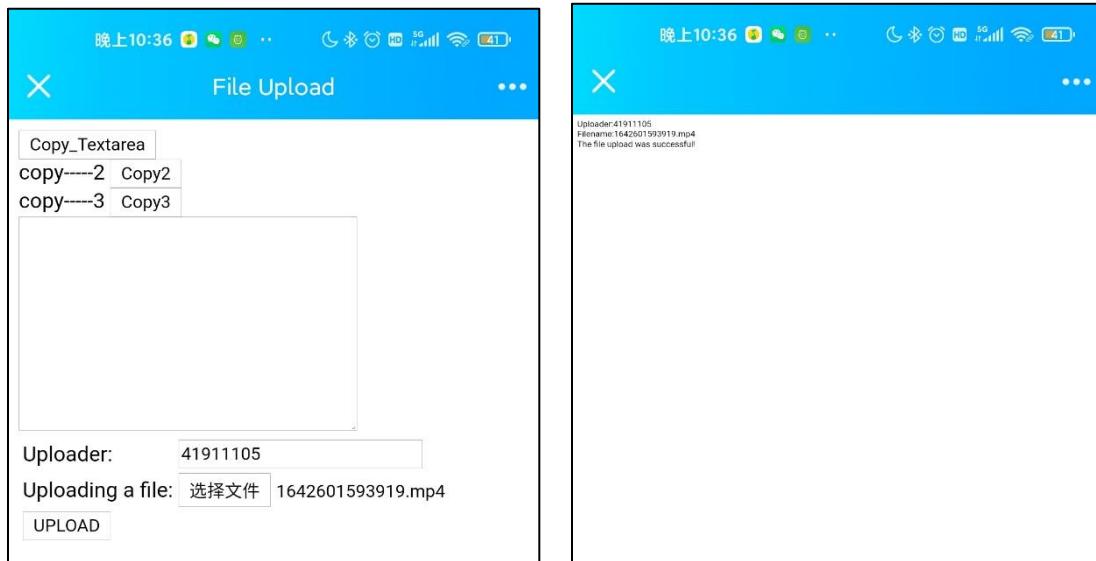


#### (2) 双击打开图片

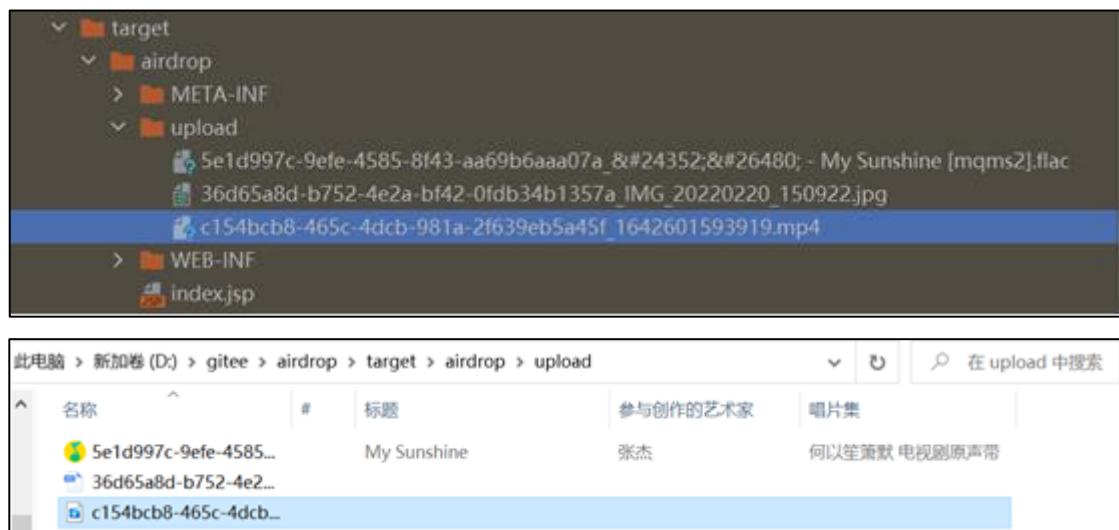


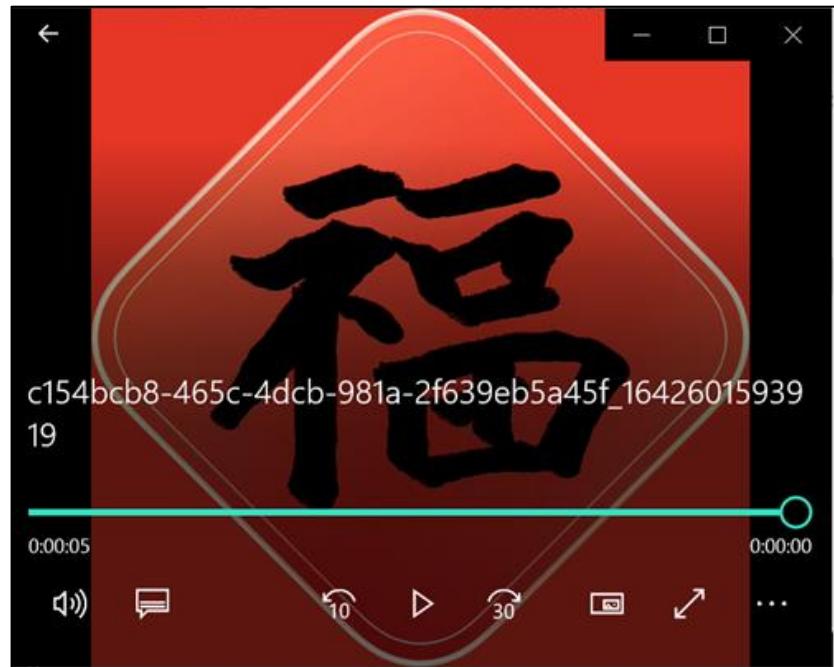
### 3.3 视频测试

#### (1) 上传视频



#### (2) 打开视频

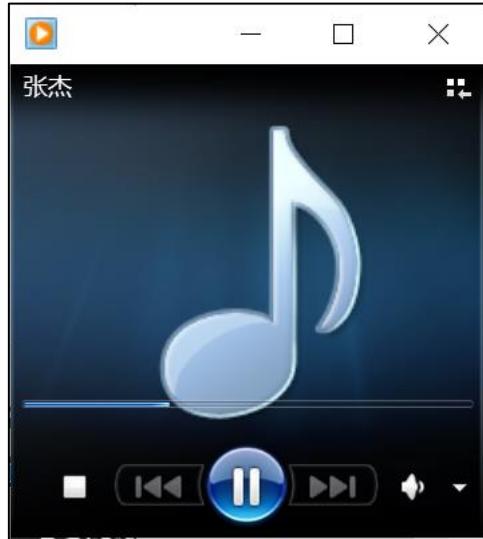




### 3.4 音乐测试

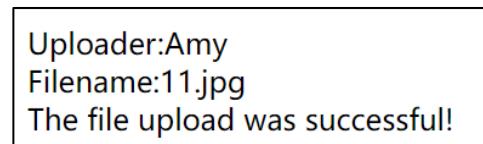
#### (1) 上传音乐

#### (2) 打开音乐

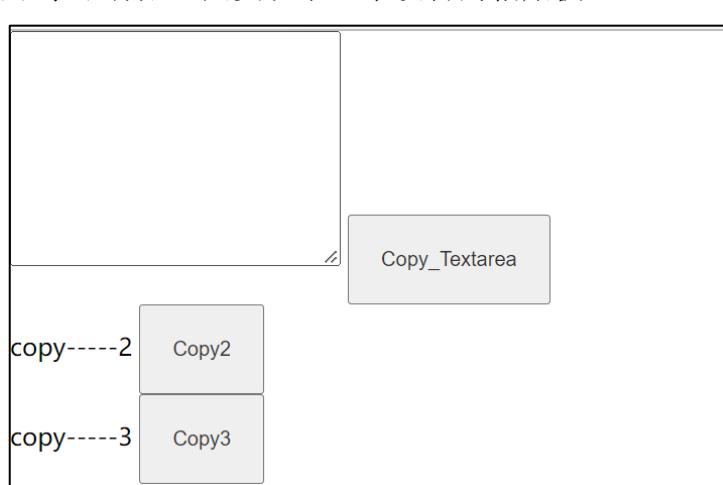


#### 4 使用教程

- (1) 打开手机热点，笔记本连接热点。使服务器（笔记本电脑）与客户端（手机）处在同一局域网内。
- (2) 在服务器（笔记本电脑）上运行项目程序。
- (3) 在手机上连接项目的 URL，输入上传者姓名并选择点击选择文件按钮，选择手机里需要传输的文件，点击 UPLOAD 按钮上传文件。上传成功后将显示上传者、文件名、成功信息。



- (4) 在服务器的项目发布目录 D:\gitee\airdrop\target\airdrop\upload 下保存有手机上传的文件，可点击查看。
- (5) 共享粘贴板由于水平有限，只实现了一个设备的粘贴板。



(6) 点击 Copy\_Textarea、Copy2、Copy3 分别可复制对应内容，在文本框内可粘贴。  
点击 Copy2(或 Copy3) 即复制按钮前的文字：copy-----2，点击后文字被选中，可在文本框内粘贴查看。



点击 Copy\_Textarea 即可复制文本框内的文字，比如在文本框中输入“你好”，点击按钮后文本框内的文字被选中，可在文本框内粘贴查看。

