

第2天 扩展内容-图片上传

- 图片上传

6. 图片上传

6.1 功能分析

题目的单选或复选，可以上传图片作为选项，页面通过后端提供上传图片组件来完成单个图片上传，图片暂时上传到服务端指定的目录保存。图片上传成功后，后端返回一个访问的相对路径（包含图片上传后的新名称），这个路径需要在表单最终提交时，与其他表单数据提交到服务端与题目选项数据一起保存到数据表中。

6.2 实现思路

后端需要提供一个单独的上传图片组件，然后返回给客户端一个可以保持到数据库的相对路径，另外对上传的图片要重新命名，保证图片名称不冲突。上传图片组件将放在一个名为公共控制器中，本图片上传不涉及其他操作，故不需要Service和Dao。

图片上传涉及到底层IO流的操作，本节将使用三方组件commons-fileupload（依赖commons-io）组件来完成图片上传的功能。

6.3 知识点补充

6.3.1 文件上传表单分析

- 凡是表单带上传文件，表单提交类型必须是【`enctype="multipart/form-data"`】
- 必须从底层IO流获取原始数据，然后进行数据解析，获取上传的数据
- 通过三方组件commons-fileupload（依赖commons-io）来解析IO流数据

6.3.2 commons组件

- commons-IO组件
 - IOUtils类
 - FileUtils类
- commons-fileupload组件
 - DiskFileItemFactory 类
 - ServletFileUpload 类
 - FileItem 类

6.4 实现上传组件

考虑到上传组件的复用性，在完成时考虑同时批量上传的情况。

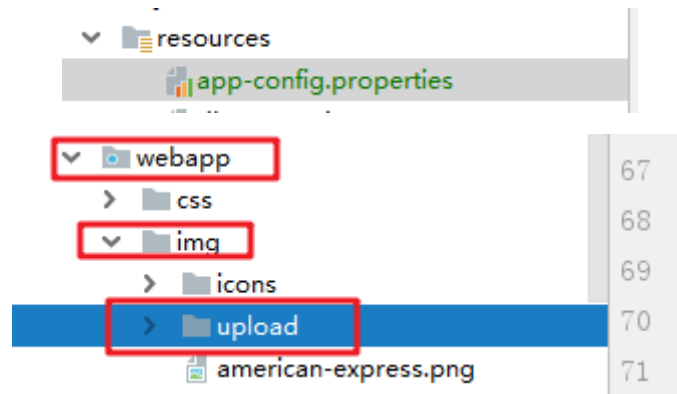
6.4.1 初始化上传文件路径配置

在resources目录下创建app-config.properties文件，该文件可以配置应用程序相关属性

```
upload_path=/img/upload
```

注意：该目录必须放置在webapps下，另外保证该目录一定初始化创建。

如图所示：



6.4.2 新增CommonController

```
/**
 * @author : seanyang
 * @date : Created in 2019/8/17
 * @description : 公共控制器
 * @version: 1.0
 */
@Component
@Slf4j
public class CommonController extends BaseController {

    @RequestMapping("/common/upload")
    public void upload(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        try {
            //创建磁盘工厂对象
            DiskFileItemFactory itemFactory = new DiskFileItemFactory();
            //创建Servlet的上传解析对象,构造方法中,传递磁盘工厂对象
            ServletFileUpload fileUpload = new ServletFileUpload(itemFactory);
            /*
             * fileUpload调用方法 parseRequest,解析request对象
             * 页面可能提交很多内容 文本框,文件,菜单,复选框 是为FileItem对象
             * 返回集合,存储的文件项对象
             */
            List<FileItem> list = fileUpload.parseRequest(request);
            List<Map<String,String>> resultList = new ArrayList<>();
            for(FileItem item : list){
                //判断普通项还是附件项
                if(!item.isFormField()){
                    // 附件项
                    // 获取上传的文件名 Lighthouse.jpg
                    String fileName = item.getName();
                    // 对文件名进行重命名,UUID+文件后缀(文件类型)
                    String newFileName = UUID.randomUUID().toString().replace("-",
                        "", "");

                    // 取文件后缀名
                    String fileType = fileName.substring(fileName.lastIndexOf('.'));
                    fileName = newFileName+fileType;
                    // 获取应用程序路径
                    String contextPath=
                        request.getSession().getServletContext().getRealPath("/");
                    log.debug("contextPath:{}", contextPath);
                    // 读取配置文件,获得上传图片的相对路径
```

```

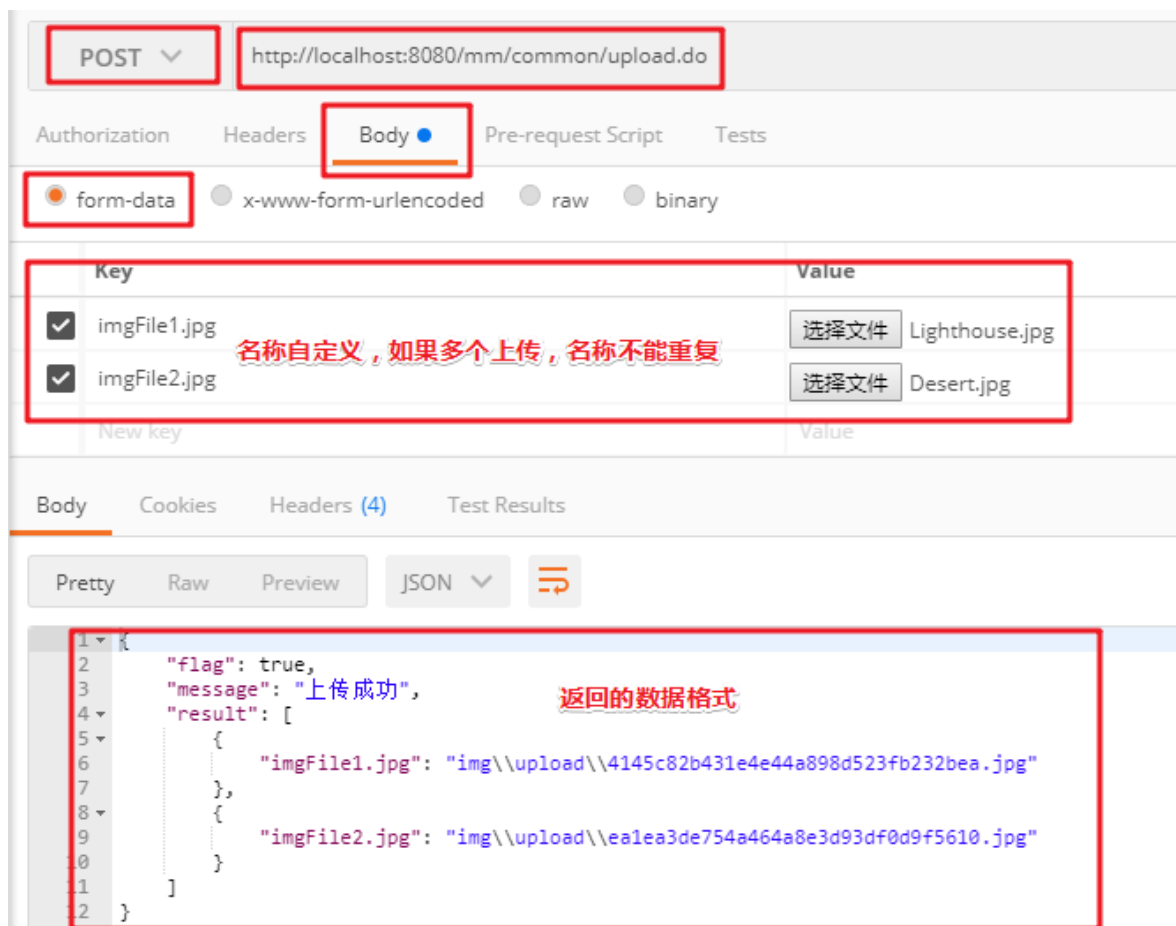
        ResourceBundle resourceBundle = ResourceBundle.getBundle("app-
config");

        String uploadDirPath= resourceBundle.getString("upload_path");
        // 读取文件数据，并存储到Web资源目录
        // img/upload
        File uploadDir = new File(contextPath,uploadDirPath);
        // 构建本地存储的文件对象
        File uploadFile = new File(uploadDir,fileName);
        // 从上传文件项获取字节输入流，然后保存到指定的上传目录下
        InputStream inputStream = item.getInputStream();
        FileOutputStream fos = new FileOutputStream(uploadFile);
        // 通过IO工具存入磁盘
        IOUtils.copy(inputStream, fos);
        // 关闭流
        IOUtils.closeQuietly(fos);
        IOUtils.closeQuietly(inputStream);
        // 存入数据库的路径
        String saveDbPath =
uploadFile.getAbsolutePath().replace(contextPath,"");
        Map mapResult = new HashMap();
        // 以上传的文件组件名为key,数据库保存的相对路径为value
        mapResult.put(item.getFieldName(),saveDbPath);
        resultList.add(mapResult);
    }
}
    printResult(response,new Result(true,"上传成功",resultList));
}catch (Exception ex){
    ex.printStackTrace();
    printResult(response,new Result(false,"上传失败,"+ex.getMessage()));
}
}
}

```

6.4.3 测试接口

在postman中，测试上传组件接口，如图所示：



6.4.4 编写前端代码

在questionEditor.html页面中，找到handleRequest方法，加入axios请求。

```

handleRequest(params, index) {
  let file = params.file;

  let formData = new FormData();
  formData.append(file.name, file);
  let response = {};
  axios.defaults.headers.post['Content-Type'] = 'multipart/form-
data;charse=UTF-8';
  axios.post(app_path + '/common/upload.do', formData).then((response) => {
    let result = response.data.result[0];
    console.log("upload-result:"+result);
    let name = Object.keys(result)[0];
    this.formData.questionItemList[index].imgUrl = result[name];
  });
}
  
```

单独测试成功后，输入表单数据，查看数据库表中的数据是否正确，正确如图所示：

id	content	img_url	is_right	question_id
116		img\upload\5b1cfd6d25cb4dc18430771d63bd8390.jpg	1	667
117		img\upload\b7d4e841605649f7a0b12c741debd16c.jpg	0	667
118		img\upload\bbf812e62a924bd09352c0a4999afd64.jpg	0	667
119		img\upload\3af2e074d13c4ab888b53f1465ad8b01.jpg	0	667

