Spring MVC

概述

> CommonsMultipartResolver是基于Apache的Commons FileUpload来实现文件上传功能的。

需要引入相关jar包：

```
<dependency>
    <groupId>commons-fileupload</groupId>
    <artifactId>commons-fileupload</artifactId>
    <version>1.3.3</version>
</dependency>
```

上传方法的实现

1. **代码方式**

```
@Controller
@RequestMapping(value = "/file")
public class FileController {

  @RequestMapping(value = "/commUploadA")
  @ResponseBody
  public JSONObject commUploadA(HttpServletRequest request) {
      JSONObject json = new JSONObject();
      json.put("succ", false);
      try {
          //直接new一个CommonsMultipartResolver
          CommonsMultipartResolver cmr = new CommonsMultipartResolver(request.getServletContext());
          cmr.setDefaultEncoding("utf-8");
          cmr.setMaxInMemorySize(40960);
          cmr.setMaxUploadSize(10485760000L);
          if (cmr.isMultipart(request)) {
              MultipartHttpServletRequest multipartRequest = cmr.resolveMultipart(request);
              MultipartFile file = multipartRequest.getFile("uploadFile");// 与页面input的name相同
              File imageFile = new File("d:/upload1.jpg");// 上传后的文件保存目录及名字
              file.transferTo(imageFile);// 将上传文件保存到相应位置
              json.put("succ", true);
              return json;
          }
      } catch (Exception e) {
          e.printStackTrace();
      }
      return json;
  }
}
```

2. **Xml方式**

> 通过spring配置一个名为“multipartResolver”的bean

```
```Xml
<bean id="multipartResolver"
```
```

```
  class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
  <property name="defaultEncoding" value="utf-8"></property>
  <property name="maxUploadSize" value="10485760000"></property>
  <property name="maxInMemorySize" value="40960"></property>
</bean>
```
controller中的方法做相应修改
```java
@Controller
@RequestMapping(value = "/file")
public class FileController {

    @RequestMapping(value = "/commUploadB")
    @ResponseBody
    public JSONObject commUploadB(MultipartHttpServletRequest request) {//参数类型不同
        JSONObject json = new JSONObject();
        json.put("succ", false);
        try {
            MultipartFile file = request.getFile("uploadFile");// 与页面input的name相同
            File imageFile = new File("d:/upload2.jpg");// 上传后的文件保存目录及名字
            file.transferTo(imageFile);// 将上传文件保存到相应位置
            json.put("succ", true);
            return json;
        } catch (Exception e) {
            e.printStackTrace();
            return json;
        }
    }
}
```

注意：

> 在使用xml方式时，bean的名字必须为：multipartResolver。可在org.springframework.web.servlet.DispatcherServlet中找到原因：DispatcherServlet#MULTIPART_RESOLVER_BEAN_NAME

Spring Boot

运用

```java
//处理文件上传
// 如果是多文件上传
@RequestMapping(value="/upload", method = RequestMethod.POST)
public @ResponseBody String uploadImg(MultipartFile file,
        HttpServletRequest request) {
    String contentType = file.getContentType();
    String fileName = file.getOriginalFilename();
    /*System.out.println("fileName-->" + fileName);
    System.out.println("getContentType-->" + contentType);*/
    String filePath = request.getSession().getServletContext().getRealPath("fileSaveDir/");
    try {
        FileUtil.uploadFile(file.getBytes(), filePath, fileName);
    } catch (Exception e) {
        // TODO: handle exception
    }
    //返回json
    return "uploadimg success";
```

```
}

// Stream写入到文件中
public static void uploadFile(byte[] file, String filePath, String fileName) throws Exception {
    File targetFile = new File(filePath);
    if(!targetFile.exists()){
        targetFile.mkdirs();
    }
    FileOutputStream out = new FileOutputStream(filePath+fileName);
    out.write(file);
    out.flush();
    out.close();
}
```

多文件上传

```
@RequestMapping(value="/upload", method = RequestMethod.POST)
public @ResponseBody String uploadImg(MultipartFile[] files,HttpServletRequest request){
    // 接收文件
}
```

文件大小控制

如果上传的文件大于 1M 时，上传会报错文件太大的错误，在 application.properties 中设置上传文件的参数即可

```
spring.http.multipart.maxFileSize=100Mb
spring.http.multipart.maxRequestSize=100Mb
```

文件下载

```
@RequestMapping(value = "/testDownload", method = RequestMethod.GET)
public void Download(HttpServletResponse res) {
  String fileName = "1.png";
  res.setHeader("content-type", "application/octet-stream");
  res.setContentType("application/octet-stream");
  res.setHeader("Content-Disposition", "attachment;filename=" + fileName);
  byte[] buff = new byte[1024];
  BufferedInputStream bis = null;
  OutputStream os = null;
  try {
    os = res.getOutputStream();
    bis = new BufferedInputStream(new FileInputStream(new File("d://" + fileName)));
    int i = bis.read(buff);
    while (i != -1) {
      os.write(buff, 0, buff.length);
      os.flush();
      i = bis.read(buff);
    }
  } catch (IOException e) {
    e.printStackTrace();
  } finally {
    if (bis != null) {
      try {
        bis.close();
      } catch (IOException e) {
        e.printStackTrace();
      }
    }
  }
```

```
  System.out.println("success");
}
```

Copy from 博客