06-SpringMVC基于注解使用：上传&下载

1、文件下载

1.1、servlet原生下载方式

```

/***
 * @Author 徐庶   QQ:1092002729
 * @Slogan 致敬大师，致敬未来的你
 */
@WebServlet("/download")
public class DownloadServlet  extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throw
        String realPath = this.getServletContext().getRealPath("/file/20181129204254948.
        File tmpFile=new File(realPath);
        String fileName = tmpFile.getName();
        response.setHeader("content-disposition", "attachment;filename="+ URLEncoder.enc
        InputStream in = new FileInputStream(realPath);//获取文件输入流
        int len = 0;
        byte[] buffer = new byte[1024];
        OutputStream out = response.getOutputStream();
        while ((len = in.read(buffer)) > 0) {
            out.write(buffer,0,len);//将缓冲区的数据输出到客户端浏览器
        }
        in.close();
    }
}
```

1.2、使用RespsonseEntity可以用来定制响应内容

```
package cn.tulingxueyuan.controller;

import cn.tulingxueyuan.bean.User;
import org.springframework.http.HttpEntity;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Controller;
import org.springframework.util.MultiValueMap;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
public class OtherController {

   @RequestMapping("/testResponseEntity")
   public ResponseEntity<String> testResponseEntity(){

       String body = "<h1>hello</h1>";
       MultiValueMap<String,String> header = new HttpHeaders();
       header.add("Set-Cookie","name=zhangsan");
       return  new ResponseEntity<String>(body,header, HttpStatus.OK);
  }
}
```

1.2、使用RespsonseEntity实现下载

```

@Controller
```

```java
public class OtherController {

    @RequestMapping("/download")
    public ResponseEntity<byte[]> download(HttpServletRequest request) throws Exception {
        //获取要下载文件的路径及输入流对象
        ServletContext servletContext = request.getServletContext();
        String realPath = servletContext.getRealPath("/script/jquery-1.9.1.min.js");
        FileInputStream fileInputStream = new FileInputStream(realPath);

        byte[] bytes = new byte[fileInputStream.available()];
        fileInputStream.read(bytes);
        fileInputStream.close();
        //将要下载文件内容返回
        HttpHeaders httpHeaders = new HttpHeaders();
        httpHeaders.set("Content-Disposition","attachment;filename=jquery-1.9.1.min.js");
        return  new ResponseEntity<byte[]>(bytes,httpHeaders,HttpStatus.OK);
    }
}

```

2、文件上传

Spring MVC 为文件上传提供了直接的支持，这种支持是通过即插即用的 **MultipartResolver** 实现的。Spring 用 **Jakarta Commons FileUpload** 技术实现了一个 MultipartResolver 实现类：**CommonsMultipartResovler**

Spring MVC 上下文中默认没有装配 MultipartResovler，因此默认情况下不能处理文件的上传工作，如果想使用 Spring 的文件上传功能，需现在上下文中配置 MultipartResolver。

pom.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xs
    <modelVersion>4.0.0</modelVersion>

    <groupId>cn.tulingxueyuan</groupId>
    <artifactId>springmvc_upload</artifactId>
    <version>1.0-SNAPSHOT</version>

    <dependencies>

```

```
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-webmvc</artifactId>
            <version>5.2.3.RELEASE</version>
        </dependency>


        <!-- https://mvnrepository.com/artifact/commons-fileupload/commons-fileupload -->
        <dependency>
            <groupId>commons-fileupload</groupId>
            <artifactId>commons-fileupload</artifactId>
            <version>1.4</version>
        </dependency>
    </dependencies>
</project>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/r
        version="4.0">

    <servlet>
        <servlet-name>springmvc</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
        <init-param>
            <param-name>contextConfigLocation</param-name>
            <param-value>classpath:springmvc.xml</param-value>
        </init-param>
    </servlet>
    <servlet-mapping>
        <servlet-name>springmvc</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>
    <filter>
        <filter-name>encoding</filter-name>
        <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-clas
        <init-param>
```

```
            <param-name>encoding</param-name>
            <param-value>UTF-8</param-value>
        </init-param>
        <init-param>
            <param-name>forceEncoding</param-name>
            <param-value>true</param-value>
        </init-param>
    </filter>
    <filter-mapping>
        <filter-name>encoding</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
</web-app>
```

springmvc.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:context="http://www.springframework.org/schema/context"
      xmlns:mvc="http://www.springframework.org/schema/mvc"

      xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springfr
    <mvc:default-servlet-handler></mvc:default-servlet-handler>
    <mvc:annotation-driven></mvc:annotation-driven>
    <context:component-scan base-package="cn.tulingxueyuan"></context:component-scan>

    <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="prefix" value="/WEB-INF/page/"></property>
        <property name="suffix" value=".jsp"></property>
    </bean>

    <bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsM
        <property name="defaultEncoding" value="UTF-8"></property>
        <property name="maxUploadSize" value="1024000"></property>
    </bean>
</beans>
```

index.jsp

```
<%--
 Created by IntelliJ IDEA.
 User: root
 Date: 2020/3/13
 Time: 17:00
 To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
 <head>
   <title>$Title$</title>
 </head>
 <body>
 <form action="testUpload" method="post" enctype="multipart/form-data">
  文件: <input type="file" name="file"/><br><br>
  描述: <input type="text" name="desc"/><br><br>
   <input type="submit" value="提交"/>
 </form>
 </body>
</html>
```

2.1、单文件上传

UploadHandler.java

```
package cn.tulingxueyuan.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.multipart.MultipartFile;

import java.io.File;
import java.io.IOException;
import java.io.InputStream;

@Controller
public class UploadHandler {

   @RequestMapping(value = "/testUpload", method = RequestMethod.POST)
   public String testUpload(@RequestParam(value = "desc", required = false) String desc,
       System.out.println("desc : " + desc);
       System.out.println("OriginalFilename : " + multipartFile.getOriginalFilename());
       multipartFile.transferTo(new File("D:\\file\\"+multipartFile.getOriginalFilename(

       return "success"; //增加成功页面: /views/success.jsp
  }
}
```

success.jsp

```
<%--
 Created by IntelliJ IDEA.
 User: root
 Date: 2020/3/13
 Time: 17:03
 To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
   <title>Title</title>
</head>
<body>
success
</body>
</html>
```

2.2、多文件上传

如果是多文件上传，那么又改如何处理呢?

index.jsp

```
<%--
 Created by IntelliJ IDEA.
 User: root
 Date: 2020/3/13
 Time: 17:00
 To change this template use File | Settings | File Templates.
```

```
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
 <head>
   <title>$Title$</title>
 </head>
 <body>
 <form action="testUpload" method="post" enctype="multipart/form-data">
  文件: <input type="file" name="file"/><br><br>
  文件: <input type="file" name="file"/><br><br>
  文件: <input type="file" name="file"/><br><br>
  描述: <input type="text" name="desc"/><br><br>
   <input type="submit" value="提交"/>
 </form>
 </body>
</html>
```

UploadHandler.java

```
/**
 * 基于springmvc MultiPartResolver多文件文件上传
 * @param desc
 * @param myfile
 * @return
 * @throws IOException
 */
@PostMapping("/upload02")
public String upload02(String desc,MultipartFile[] myfile) throws  IOException {

    for (MultipartFile multipartFile : myfile) {
        System.out.println(desc);
        System.out.println(multipartFile.getOriginalFilename());
        String path = "D:\\out\\" + multipartFile.getOriginalFilename();
        File file = new File(path);
        multipartFile.transferTo(file);
    }
    return "success";
}

```

2.3、多文件多线程上传

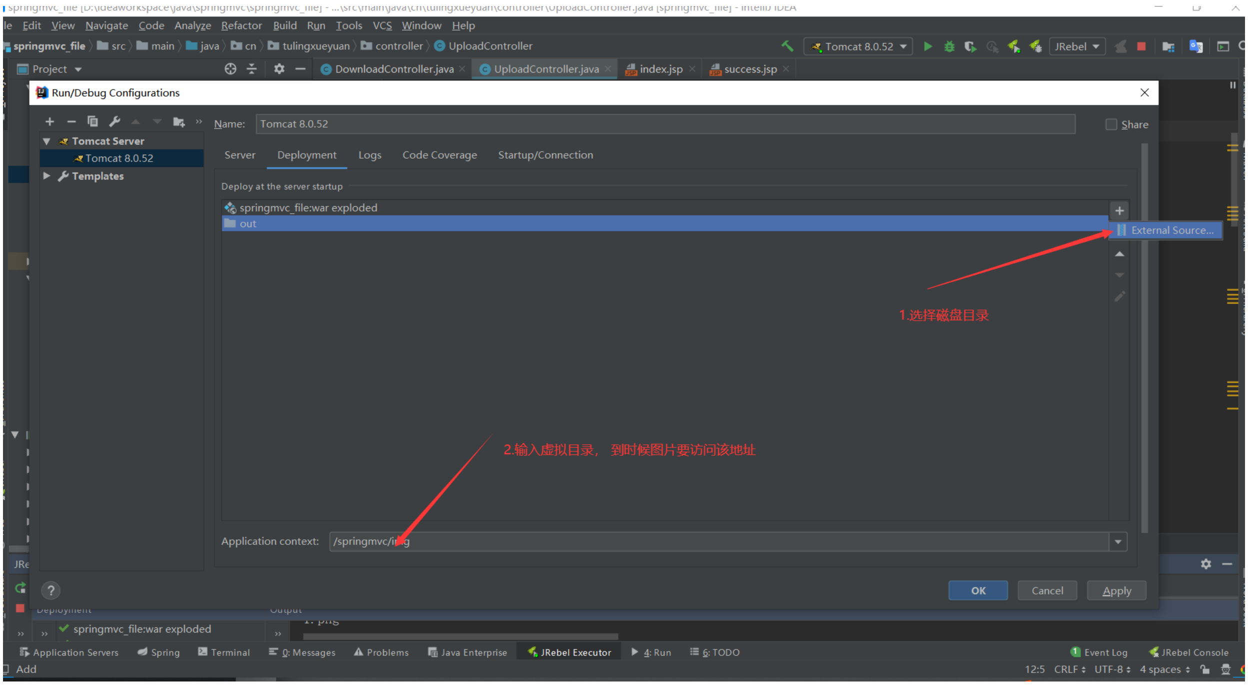
考虑到多文件的性能，多线程文件上传可以利用多线程的非阻塞特点来实现，性能更加高效：

```
/**
 * 基于springmvc MultiPartResolver多文件文件上传--多线程
 * @param desc
 * @param myfile
 * @return
 * @throws IOException
 */
@PostMapping("/upload03")
public String upload03(String desc,MultipartFile[] myfile) throws IOException, Interrupte

    System.out.println(desc);
    for (MultipartFile multipartFile : myfile) {
        // 声明线程
        Thread thread = new Thread(() -> {
            System.out.println(multipartFile.getOriginalFilename());
            String path = "D:\\out\\" + multipartFile.getOriginalFilename();
            File file = new File(path);
            try {
                multipartFile.transferTo(file);
            } catch (IOException e) {
                e.printStackTrace();
            }
        });
        thread.start();   //启动线程
        thread.join();   // 让子线程执行完再执行主线程
    }
    return "success";
}
```

2.4、上传磁盘路径显示图片：

1.映射虚拟目录



Name: Tomcat 8.0.52

Deploy at the server startup

- springmvc_file:war exploded
- out

1.选择磁盘目录

2.输入虚拟目录， 到时候图片要访问该地址

Application context: /springmvc/img

2.直接显示

```
/**
 * 基于springmvc MultiPartResolver文件上传
 * @param desc
 * @param multipartFile
 * @return
 * @throws IOException
 */
@PostMapping("/upload01")
public String upload01(String desc, @RequestParam("myfile") MultipartFile multipartFile,

    System.out.println(desc);
    System.out.println(multipartFile.getOriginalFilename());
    String path = "D:\\out\\" + multipartFile.getOriginalFilename();
    File file = new File(path);
    multipartFile.transferTo(file);
    // 上传后添加文件名到域中
    model.addAttribute("filename",multipartFile.getOriginalFilename());
    return "success";
}
```

jsp:

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
```

```
<head>
    <title>Title</title>
</head>
<body>
上传成功!
<img src="${pageContext.request.contextPath}/img/${filename}">
</body>
</html>
```