springboot(十七)：使用Spring Boot上传文件

 2018/01/12

上传文件是互联网中常常应用的场景之一，最典型的情况就是上传头像等，今天就带着带着大家做一个Spring Boot上传文件的小案例。

**1、pom包配置**

我们使用Spring Boot最新版本1.5.9、jdk使用1.8、tomcat8.0。

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>1.5.9.RELEASE</version>

</parent>

<properties>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-thymeleaf</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<optional>true</optional>

</dependency>

</dependencies>

引入了spring-boot-starter-thymeleaf做页面模板引擎，写一些简单的上传示例。

**2、启动类设置**

**@SpringBootApplication**

**public** **class** **FileUploadWebApplication** **{**

**public** **static** **void** **main(**String**[]** args**)** **throws** Exception **{**

SpringApplication**.**run**(**FileUploadWebApplication**.**class**,** args**);**

**}**

*//Tomcat large file upload connection reset*

**@Bean**

**public** TomcatEmbeddedServletContainerFactory **tomcatEmbedded()** **{**

TomcatEmbeddedServletContainerFactory tomcat **=** **new** TomcatEmbeddedServletContainerFactory**();**

tomcat**.**addConnectorCustomizers**((**TomcatConnectorCustomizer**)** connector **->** **{**

**if** **((**connector**.**getProtocolHandler**()** **instanceof** AbstractHttp11Protocol**<?>))** **{**

*//-1 means unlimited*

**((**AbstractHttp11Protocol**<?>)** connector**.**getProtocolHandler**()).**setMaxSwallowSize**(-**1**);**

**}**

**});**

**return** tomcat**;**

**}**

**}**

tomcatEmbedded这段代码是为了解决，上传文件大于10M出现连接重置的问题。此异常内容GlobalException也捕获不到。



详细内容参考：[Tomcat large file upload connection reset](http://www.mkyong.com/spring/spring-file-upload-and-connection-reset-issue/)

**3、编写前端页面**

上传页面

**<!DOCTYPE html>**

<html xmlns:th="http://www.thymeleaf.org">

<body>

<h1>Spring Boot file upload example</h1>

<form method="POST" action="/upload" enctype="multipart/form-data">

<input type="file" name="file" /><br/><br/>

<input type="submit" value="Submit" />

</form>

</body>

</html>

非常简单的一个Post请求，一个选择框选择文件，一个提交按钮，效果如下：



上传结果展示页面：

**<!DOCTYPE html>**

<html lang="en" xmlns:th="http://www.thymeleaf.org">

<body>

<h1>Spring Boot - Upload Status</h1>

<div th:if="${message}">

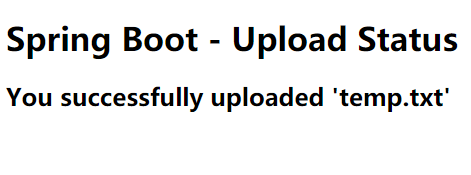
<h2 th:text="${message}"/>

</div>

</body>

</html>

效果图如下：



**4、编写上传控制类**

访问localhost自动跳转到上传页面：

**@GetMapping(**"/"**)**

**public** String **index()** **{**

**return** "upload"**;**

**}**

上传业务处理

**@PostMapping(**"/upload"**)**

**public** String **singleFileUpload(@RequestParam(**"file"**)** MultipartFile file**,**

RedirectAttributes redirectAttributes**)** **{**

**if** **(**file**.**isEmpty**())** **{**

redirectAttributes**.**addFlashAttribute**(**"message"**,** "Please select a file to upload"**);**

**return** "redirect:uploadStatus"**;**

**}**

**try** **{**

*// Get the file and save it somewhere*

**byte[]** bytes **=** file**.**getBytes**();**

Path path **=** Paths**.**get**(**UPLOADED\_FOLDER **+** file**.**getOriginalFilename**());**

Files**.**write**(**path**,** bytes**);**

redirectAttributes**.**addFlashAttribute**(**"message"**,**

"You successfully uploaded '" **+** file**.**getOriginalFilename**()** **+** "'"**);**

**}** **catch** **(**IOException e**)** **{**

e**.**printStackTrace**();**

**}**

**return** "redirect:/uploadStatus"**;**

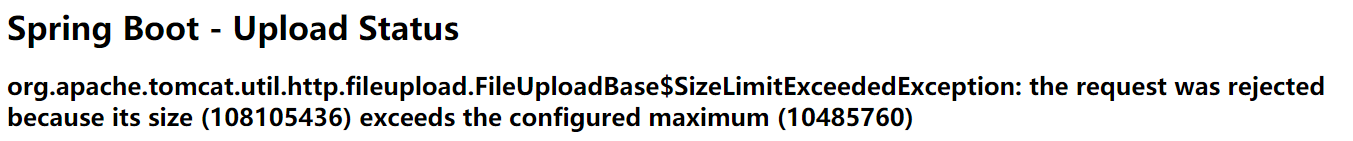
**}**

上面代码的意思就是，通过MultipartFile读取文件信息，如果文件为空跳转到结果页并给出提示；如果不为空读取文件流并写入到指定目录，最后将结果展示到页面。

MultipartFile是Spring上传文件的封装类，包含了文件的二进制流和文件属性等信息，在配置文件中也可对相关属性进行配置，基本的配置信息如下：

* spring.http.multipart.enabled=true #默认支持文件上传.
* spring.http.multipart.file-size-threshold=0 #支持文件写入磁盘.
* spring.http.multipart.location= # 上传文件的临时目录
* spring.http.multipart.max-file-size=1Mb # 最大支持文件大小
* spring.http.multipart.max-request-size=10Mb # 最大支持请求大小

最常用的是最后两个配置内容，限制文件上传大小，上传时超过大小会抛出异常：



更多配置信息参考这里：[Common application properties](https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#common-application-properties)

**5、异常处理**

**@ControllerAdvice**

**public** **class** **GlobalExceptionHandler** **{**

**@ExceptionHandler(**MultipartException**.**class**)**

**public** String **handleError1(**MultipartException e**,** RedirectAttributes redirectAttributes**)** **{**

redirectAttributes**.**addFlashAttribute**(**"message"**,** e**.**getCause**().**getMessage**());**

**return** "redirect:/uploadStatus"**;**

**}**

**}**

设置一个@ControllerAdvice用来监控Multipart上传的文件大小是否受限，当出现此异常时在前端页面给出提示。利用@ControllerAdvice可以做很多东西，比如全局的统一异常处理等，感兴趣的同学可以下来了解。

**6、总结**

这样一个使用Spring Boot上传文件的简单Demo就完成了，感兴趣的同学可以将示例代码下载下来试试吧。

**参考**：

[Spring Boot file upload example](http://www.mkyong.com/spring-boot/spring-boot-file-upload-example/)

[**示例代码-github**](https://github.com/ityouknow/spring-boot-examples)

[**示例代码-码云**](https://gitee.com/ityouknow/spring-boot-examples)