

1. 声明MultipartResolver

spring默认支持两种上传方式，一个是基于 `Servlet3.0` 的 `StandardServletMultipartResolver`，一个是基于 `commons` 的 `CommonsMultipartResolver`。在spring-boot中，会默认帮我们声明 `StandardServletMultipartResolver`，要想更改成 `commons`，需要在spring-boot中包含主函数的类中修改成如下代码：

```
1.  @Configuration
2.  //exclude表示自动配置时不包括Multipart配置
3.  @EnableAutoConfiguration(exclude={MultipartAutoConfiguration.class})
4.  public class UploadApplication extends WebMvcConfigurerAdapter {
5.
6.      public static void main(String[] args) {
7.          SpringApplication.run(UploadApplication.class, args);
8.      }
9.
10.     //显式声明CommonsMultipartResolver为mutipartResolver
11.     @Bean(name="multipartResolver")
12.     public MultipartResolver mutipartResolver() {
13.         CommonsMultipartResolver com = new CommonsMultipartResolver();
14.         com.setDefaultEncoding("utf-8");
15.         return com;
16.     }
17. }
```

2. 使用MultipartResolver

若表单的提交方式为 `multipart/form-data`，spring会将request根据声明的 `mutipartResolver` 转换为对应的 `MultipartHttpServletRequest`，我们也可以在代码中这样做：

```
1.  MultipartHttpServletRequest mRequest
2.  = (MultipartHttpServletRequest) request;
```

如此，我们就能使用 `mRequest` 得到上传文件：

```
1.  MultipartFile getFile(String name);           或
2.  List<MultipartFile> getFiles(String name);   或
3.  Map<String, MultipartFile> getFileMap();
```

得到的类型为 `MultipartFile`，里面封装了File的各种信息。

3. 最后

综上，作为spring的新项目，spring-boot还是很人性化的，它确实为项目开发提供了更加快速简洁的方法。

-END-