链滴

springMVC 获取本地项目路径 以及上传文件的方法整理

作者：huihui

原文链接：https://ld246.com/article/1482216411523

来源网站：链滴

SpringMVC是一个基于DispatcherServlet的MVC框架，每一个请求最先访问的都是 DispatcherServlet，DispatcherServlet负责转发每一个Request请求给相应的Handler，Handler处理 以后再返回相的视图(View)和模型(Model)，返回的视图和模型都可以不指定，即可以只返回Model或只返回View都不返回。在使用注解 的SpringMVC中，处理器Handler是基于@Controller和@RequestMappin这两个注解的，@Controller声明 一个处理器类，@RequestMapping声明对应请求的映射关系，这就可以提供一个非常灵活的匹配和处理方式。

DispatcherServlet是继承自HttpServlet的，既然SpringMVC是基于DispatcherServlet的，那么 我先来配置一下DispatcherServlet，好让它能够管理我们希望它管理的内容。HttpServlet是在web.xm文件中声明的。

一、从视图向controller传递值， controller <— 视图

1、通过@PathVariabl注解获取路径中传递参数

```
1    @RequestMapping(value = "/{id}/{str}")
2    public ModelAndView helloWorld(@PathVariable String id,
3            @PathVariable String str) {
4        System.out.println(id);
5        System.out.println(str);
6        return new ModelAndView("/helloWorld");
7    }
```

2、

1）简单类型，如int, String, 应在变量名前加@RequestParam注解，

例如：

```
@RequestMapping("hello3")
public String hello3( @RequestParam("name" ) String name,
                    @RequestParam("hobby" ) String hobby){
    System. out.println("name=" +name);
    System. out.println("hobby=" +hobby);
     return "hello" ;
}
```

但这样就要求输入里面必须有这两个参数了，可以用required=false来取消，例如：

@RequestParam(value=”name”,required=false) String name

但经测试也可以完全不写这些注解，即方法的参数写String name，效果与上面相同。

2）对象类型：

```
@RequestMapping("/hello4" )
public String hello4(User user){
    System.out.println("user.getName()=" +user.getName());
    System.out.println("user.getHobby()=" +user.getHobby());
    return "hello";
}
```

Spring MVC会按：

“HTTP请求参数名= 命令/表单对象的属性名”

的规则自动绑定请求数据，支持“级联属性名”，自动进行基本类型数据转换。

此外，还可以限定提交方法为POST，即修改方法的@RequestMapping注解为

@RequestMapping(value=”/hello4”,method=RequestMethod.POST)

最后，注意，如果这里提交过来的字符出现乱码，应该在web.xml里加入如下filter:

encodingFilter

org.springframework.web.filter.CharacterEncodingFilter

```
 encoding
 utf8
```

encodingFilter

/*

返回数据到页面几种方式：

1.

```
//返回页面参数的第二种方式,在形参中放入一个Model
@RequestMapping(value = "/hello2.htm")
public String hello2(int id,Model model){
    System.out.println("hello2 action:"+id);
    model.addAttribute("name", "huangjie");
    //这个只有值没有键的情况下,使用Object的类型作为key,String-->string
    model.addAttribute("ok");
    return "hello";
}
```

2.

```
//返回页面参数的第一种方式,在形参中放入一个map
@RequestMapping(value = "/hello1.htm")
public String hello(int id,Map map){
    System.out.println("hello1 action:"+id);
    map.put("name", "huangjie");
    return "hello";
}
```