# Spring MVC

Spring MVC 注解类型的优点

(1) 一个控制器可以处理多个动作，允许将相关操作写在同一个控制器内。

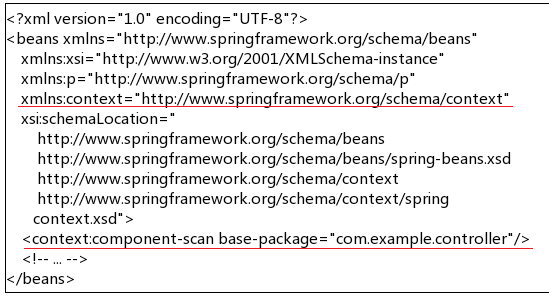
1. 基于注解的控制器的请求映射不需要存储在配置文件中，使用RequestMapping注释类型，可以对一个方法进行请求处理。

**org.springframework.stereotype.Controller**注解类型用于指示Spring类的实例是一个**控制器**。为了保证Spring能找到控制器，就需要

1, 需要在SpringMVC的配置文件中声明spring-context

2, 需要指明<component-scan />元素

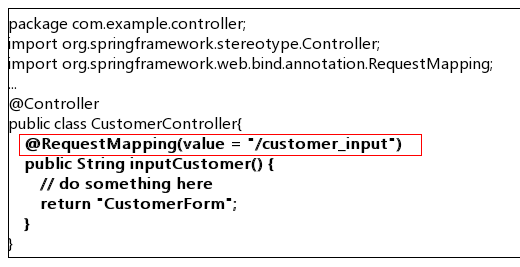
如图



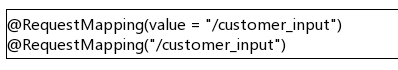
**org.springframework.web.bind.annotation.RequestMapping**注释类型用于映射一个请求和一个方法，可以使用@RequestMapping注释一种方法和类。

一个采用@RequestMapping注释的方法将成为一个请求处理方法，并由调度程序在接收到对应URL请求时调用。RequestMapping注解的value属性值将URL映射到方法。

<http://domain/context>/customer\_input

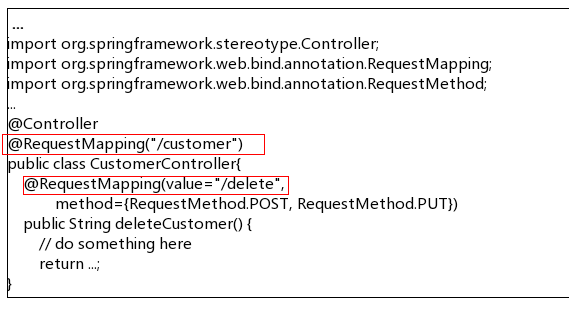


value属性是RequestMapping注释的默认属性，若只有唯一的属性时可以省略属性名称。若超过一个属性时，就必须写入value属性名称。请求映射的value值可以是一个空字符串，此时该方法被映射到 <http://domain/context> 网址。



RequestMapping注释类型也可以用来注释一个控制器(类)。此时，所有的方法都将映射为相对于类级别的请求。

<http://domain/context/customer/delete>



RequestMapping请求处理方法参数支持的**参数类型**：

javax.servlet.ServletRequest

javax.servlet.http.HttpServletRequest

javax.servlet.ServletResponse javax.servlet.http.HttpServletResponse

javax.servlet.http.HttpSession

org.springframework.web.context.request.WebRequest

org.springframework.web.context.request.NativeWebRequest

java.util.Locale

java.io.InputStream

java.io.Reader

java.io.OutputStream

java.io.Writer

java.security.Principal

HttpEntity<?>

java.util.Map

org.springframework.ui.Model

org.springframework.ui.ModelMap

org.springframework.web.servlet.mvc.support.RedirectAttributes

org.springframework.validation.Errors

org.springframework.validation.BindingResult

命令或表单对象

org.springframework.web.bind.support.SessionStatus

org.springframework.web.util.UriComponentsBuilder

带@PathVariable, @MatrixVariable注释的对象

@RequestParam, @RequestHeader, @RequestBody，@RequestPart

请求处理方法支持的返回对象类型：

ModelAndView

Model

Map包含模型的属性

View

代表逻辑视图名的String

Void

提供对Servlet的访问，以响应HTTP头部和内容HttpEntity或ResponseEntity对象

Callable

DeferredResult

其他任意类型，Spring将其视作输出给View的对象模

**@RestController** 与 **@Controller**的区别

@RestController的编写方式依赖注解组合，@RestController被@Controller和@ResponseBody标注，表示@RestController具有两者的注解语义，因此在注解处理时@RestController比@Controller多具有一个@ResponseBody语义，这就是@RestController和@Controller的区别，也是@RestController的返回值为何都是经过转换的json的原因。

所以小结就是：@RestController = @Controller + @ResponseBody