**Spring中参数绑定**非常灵活，他会把数据类型不匹配当成验证错误，而非系统错误。

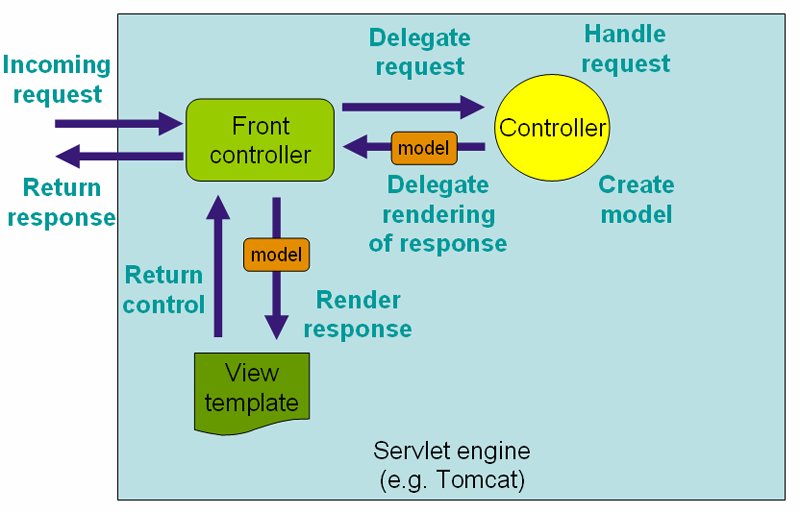
不必为了非法类型转换在表单对象中使用字符串来存储数据。

WebApplicationContext spring web 模块中的类、而ApplicationContext是context中的类。

WebApplicationContext（是指容器上下文对象，是beanfactory的子类 ）被绑定在ServletContext中。如果需要获取它，你可以通过RequestContextUtils工具类中的静态方法来拿到这个web应用的上下文WebApplicationContext。

**架构：**则包含代码、设计、分析的重用

**MVC框架：**框架代码的重用包括spring web mvc struts



**MVC设计模式（并不在23中设计模式中，是软件设计模式理念）：**设计的重用



Springmvc 设计围绕dispatcherservlet中央servlet，负责将所有请求分发到控制器，实际上是一个servlet。

WebApplicationContext中一些特殊的bean类型：

Handlermapping 处理器映射，将请求映射到指定处理器

HandlerAdapter 处理器适配器，找到处理器后负责调用该处理器，分离中央servlet和具体实现的耦合。

[HandlerExceptionResolver](http://7xvpsh.com1.z0.glb.clouddn.com/publish/21-11/1-handler-exception-handler.html) 负责将捕捉的异常映射到不同的视图上去

ViewResolver视图解析器。它负责将一个代表逻辑视图名的字符串（String）映射到实际的视图类型View上。

**@requestmapper**

可以定义在类上和方法上

Value设置映射路径

Path 4.2版本后新增，和value一样均用来指定映射路径

在映射路径中请求优先访问最准确的路径

占位符的路径模式${…}取得一些本地配置、系统配置、环境变量

Method 限制请求的方法

Consumes 消费的媒体类型，即限制请求头中content-type需要包含在consumes中。

Produces 生产的媒体类型，限制请求头中的accept（请求中accept默认为全部类型）的类型；指定响应头content-type类型。浏览器会根据该类型解析返回的数据。

Params：指定请求参数以及参数值支持post和get方式

Heads：指定请求头，consumes和produces是heads的特例。

**url模板**

类似uri，只不过包含了一个或多个变量名，请求过来时会根据url去填充对应位置变量，并且可通过@pathvariable注解该值传入方法参数中。

**@Requestmapping注解的方法：**

**支持的方法参数类型：**

<http://itindex.net/detail/50836-%E6%8E%A7%E5%88%B6%E5%99%A8-springmvc-%E6%95%B0%E6%8D%AE>（详情网址）

1. 请求或响应对象，如：ServletRequest或HttpServletRequest对象；Httpsession类型的回话对象（session对象存在线程安全问题，存取session可能不是线程安全的，特别是在一个Servlet的运行环境中。如果应用可能有多个请求同时并发存取一个session场景，请考虑将RequestMappingHandlerAdapter类中的"synchronizeOnSession"标志设置为"true"。）；
2. 当前请求的地区信息java.util.Locale；
3. WebRequest/NativeWebRequest Spring Web MVC提供的统一请求访问接口，不仅仅可以访问请求相关数据；
4. 与当前请求绑定的时区信息java.util.TimeZone
5. 存取请求的输入流（java.io.InputStream或java.io.Reader），生成相应的输出流(java.io.OutputStream或java.io.Writer)
6. HttpMethod http请求方法
7. @PathVariable注解的参数，是指url模板中变量的值
8. @MatrixVarible注解的参数，矩阵变量（url重写后面携带的变量）
9. @requestparam注解的参数， servlet请求中指定的参数，参数值转换成方法参数声明的类型。主要用于由浏览器提交表单请求（包括文件，基本类型数据）
10. @requestbody 注解的参数，请求体中的参数，参数是一个json串
11. @requestpart 注解的参数，客户端请求提交的复杂参数类型，如非浏览器的远程调用接口，通过post传输，请求体中包括复杂数据类型。
12. HttpEntity<?>类型的参数其提供对http请求头和内容的存取，请求流是通过HttpMessageConverter被转换成entity对象的，替换原生HttpServletRequest。
13. Map、model、modelmap 增强默认暴露给视图层的模型
14. org.springframework.web.servlet.mvc.support.RedirectAttributes类型的参数，用于指定重定向的参数。
15. 命令或表单对象
16. org.springframework.validation.Errors/org.springframework.validation.BindingResult验证结果对象，用于存储前面的命令或表单对象的验证结果（紧接其前的第一个方法参数）注意：该参数必须要紧跟在校验对象后。
17. org.springframework.web.util.UriComponentsBuilder 构造器对象，用于构造当前请求URL相关的信息，比如主机名、端口号、资源类型（scheme）、上下文路径、servlet映射中的相对部分（literal part）等。

**支持的方法返回值类型：**

1. modelandview、model、map、view对象，包含model隐含填充的命令对象以及modelattribute注解的字段
2. String对象，其值默认被解析为一个 逻辑视图名，model将默认填充了命令对象以及注解了 @ModelAttribute 字段的存取器被调用所返回的值
3. Void，当方法已对响应进行了处理。
4. 处理器方法注解了ResponseBody，返回类型会被写进http的响应体（将被写入HTTP的响应流中（直接输出在页面），而不会被被放置到Model中或被解释为一个视图名），返回值也会转换成方法声明的返回值类型。
5. Httpentity或ResponseEntity用于提供对Servlet HTTP响应头和响应内容的存取
6. HttpHeaders 返回一个不包含响应体的响应
7. 异步返回对象Callable<?>
8. 其他任何返回类型，都会被处理成model的一个属性并返回给视图，该属性的名称为方法级的 @ModelAttribute 所注解的字段名（或者以返回类型的类名作为默认的属性名）。model隐含填充了命令对象以及注解了 @ModelAttribute 字段的存取器被调用所返回的值

**@requestparam**：请求参数绑定，注解的参数类型是 Map<String, String>时会自动填充所有参数。

**@requestbody：**请求体绑定。文档演示为put请求。暂时未知

**@ResponseBody** 注解可被应用于方法上，将返回值写入响应体中。

**@modelattribute** 可注解在方法或方法参数上

方法上：

单独注解在方法：

注解了 @ModelAttribute 的方法实际上会在 @RequestMapping 方法之前被调用，通常被用来填充一些公共需要的属性或数据，如下两种代码风格，第一种默认将返回值添加到model中，model中属性名默认为类型名去首字母大写。第二种在model中手动添加属性。

@modelattribute和@requestmapping一同注解方法时，返回值会会被解释为model中的属性而部署一个视图名。

|  |
| --- |
| @ModelAttribute  public Account addAccount(@RequestParam String number) {  return accountManager.findAccount(number);  }  @ModelAttribute  public void populateModel(@RequestParam String number, Model model) {  model.addAttribute(accountManager.findAccount(number));  } |

方法参数上：

model中找不到，那么该参数会先被实例化，然后被添加到model中。在model中存在以后，请求中所有名称匹配的参数都会填充到该参数中。这在Spring MVC中被称为数据绑定（），一个非常有用的特性，节约了你每次都需要手动从表格数据中转换这些字段数据的时间。

注解@modelattribute的参数实例化可能来源：

1. 如果方法之前使用了@sessionattribute，可能已经存在于model中了
2. 可能通过url模板中的变量获得
3. 调用自身默认构造器被实例化出来的。

数据绑定过程中数据的处理则通过webdataBinder来进行的

ServletRequest.getParameter\*() 系列的方法只能支持通过HTTP POST方法的方式提交表单，而不支持HTTP PUT的方式。

Springmvc对于put请求的支持，通过过滤器中获取数据并封装在request中。

**方法参数与类型转换**

如使用@modelattribute注解的参数，如果目标参数与方法参数类型不一致时，将要进行强转，在这里使用webdataBinder处理，通过@initbinder注解方法实现。