## Springmvc参数绑定过程

从客户端请求key/value数据，经过参数绑定，将key/value数据绑定到controller方法的形参上。

**Springmvc中，接收页面提交的数据是通过方法形参来接收**

处理器适配器调用springmvc提供参数绑定组件将key/value数据转成controller方法的形参

参数绑定组件：在spirngmvc早期版本使用PropertyEditor（只能将字符串传成java对象）

后期使用converter（进行任意类型的传换）

spirngmvc提供了很多converter（转换器）

在特殊情况下需要自定义converter

对日期数据绑定需要自定义converter

## 默认支持的类型

直接在controller方法形参上定义下边类型的对象，就可以使用这些对象。在参数绑定过程中，如果遇到下边类型直接进行绑定。

HttpServletRequest

通过request对象获取请求信息

HttpServletResponse

通过response处理响应信息

HttpSession

通过session对象得到session中存放的对象

Model/ModelMap

model是一个接口，modelMap是一个接口实现 。

作用：将model数据填充到request域。

### 简单类型

通过**@RequestParam**对简单类型的参数进行绑定。

**如果不使用@RequestParam，要求request传入参数名称和controller方法的形参名称一致，方可绑定成功。**

**如果使用@RequestParam，不用限制request传入参数名称和controller方法的形参名称一致。**

通过required属性指定参数是否必须要传入，如果设置为true，没有传入参数，报下边错误：



### 简单pojo(属性为简单类型)绑定

页面中input的name和controller的pojo形参中的属性名称一致，将页面中数据绑定到pojo。

页面定义：

将pojo对象中的属性名于传递进来的属性名对应，如果传进来的参数名称和对象中的属性名称一致则将参数值设置在pojo对象中

页面定义如下;

<input type="text" name="name"/>

<input type="text" name="price"/>

Contrller方法定义如下：

@RequestMapping("/editItemSubmit")

**public** String editItemSubmit(Items items)**throws** Exception{

System.out.println(items);

请求的参数名称和pojo的属性名称一致，会自动将请求参数赋值给pojo的属性。

### 包装pojo（属性为复合类型）绑定

如果采用类似struts中对象.属性的方式命名，需要将pojo对象作为一个包装对象的属性，action中以该包装对象作为形参。

包装对象定义如下：

**Public class** QueryVo {

**private** Items items;

}

页面定义：

<input type="text" name="items.name" />

<input type="text" name="items.price" />

Controller方法定义如下：

public String useraddsubmit(Model model,QueryVo queryVo)throws Exception{

System.out.println(queryVo.getItems());