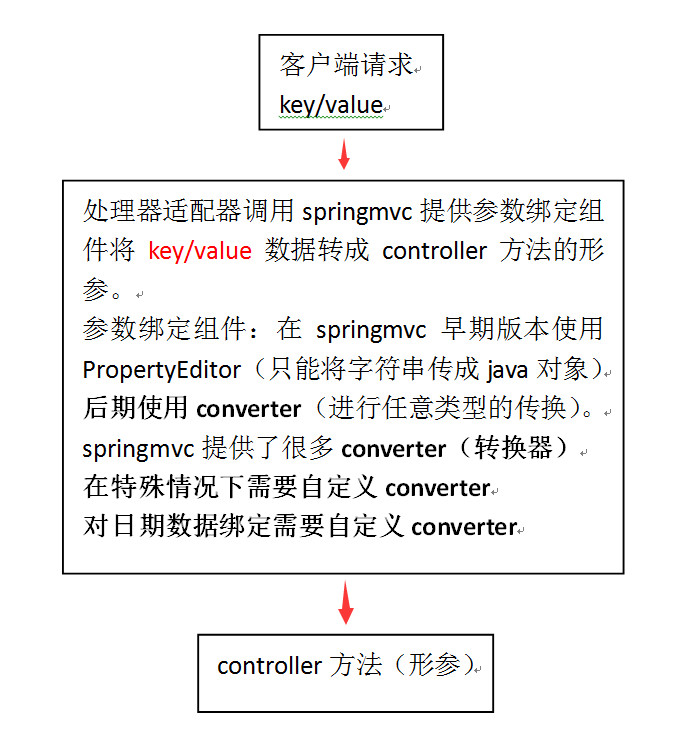
**[SpringMVC中的参数绑定总结](https://www.cnblogs.com/linjiaxin/p/7753374.html)**

## ****1. 参数绑定的过程****

　　我们可以回忆一下，在struts2中，是通过在Action中定义一个成员变量来接收前台传进来的参数，而在springmvc中，接收页面提交的数据是通过方法形参来接收的。从客户端请求的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，然后就可以在controller中使用该参数了。来看一下这个过程



所以我们知道，是springmvc提供了很多转换器来将页面参数绑定到controller方法的形参上，关于自定义converter，我下面会提到。大概了解了该过程后，下面开始做具体的总结。

## ****2. 默认支持的类型****

　　springmvc中，有支持的默认类型的绑定。也就是说，直接在controller方法形参上定义默认类型的对象，就可以使用这些对象。

1. HttpServletRequest对象
2. HttpServletResponse对象
3. HttpSession对象
4. Model/ModelMap对象

　　在参数绑定过程中，如果遇到上面类型就直接进行绑定。也就是说，我们可以在controller的方法的形参中直接定义上面这些类型的参数，springmvc会自动绑定。这里要说明一下的就是Model/ModelMap对象，Model是一个接口，ModelMap是一个接口实现 ，作用是将Model数据填充到request域，跟ModelAndView类似，关于它的使用，我在下面和简单类型参数绑定一起说。

## ****3. 简单类型的绑定****

　　总结这个还是以需求为例吧，这样比较容易理解，假设现在有个需求：根据商品的id来修改对应点商品信息。所以前台页面肯定要传进来该商品的id，然后springmvc的controller进行处理，返回一个修改商品信息的页面。关于前台页面的东西都很简单，我就不贴代码了，主要部分截个图，具体的代码在文章最后有下载地址。   
　　前台页面通过url将参数传递过来，请求的是editItems.action。   
简单类型   
　　下面写controller中的editItems方法：

@RequestMapping("/editItems")

public String editItems(Model model, Integer id) throws Exception {

//根据id查询对应的Items

ItemsCustom itemsCustom = itemsService.findItemsById(id);

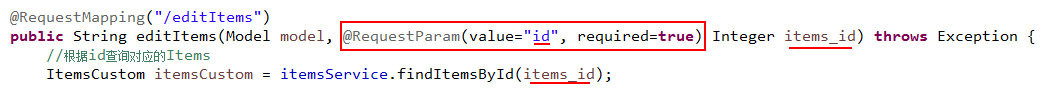
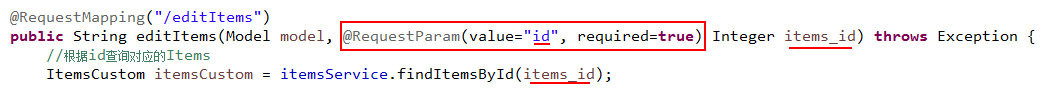
model.addAttribute("itemsCustom", itemsCustom);

//通过形参中的model将model数据传到页面

//相当于modelAndView.addObject方法

return "/WEB-INF/jsp/items/editItems.jsp";

}

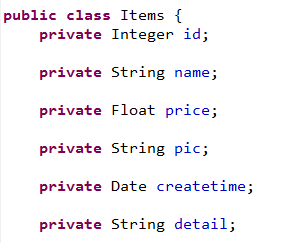
这是个很简单的demo，从上面的代码中可以看出model可以直接作为参数，springmvc默认会绑定它，然后使用model将查询到的数据放到request域中，这样就可以在前台页面取出该数据了。   
　　要注意一点的是，简单类型的绑定中，方法形参中的参数名要和前台传进来的名一样才能完成参数的绑定。那有人要问了，如果有特殊需求（比如更好的可读性？），这里定义的参数名就是不一样，那咋整呢？有解决办法么？有！我们可以使用注解@RequestParam对简单的类型进行参数绑定，如下：   
  
　　

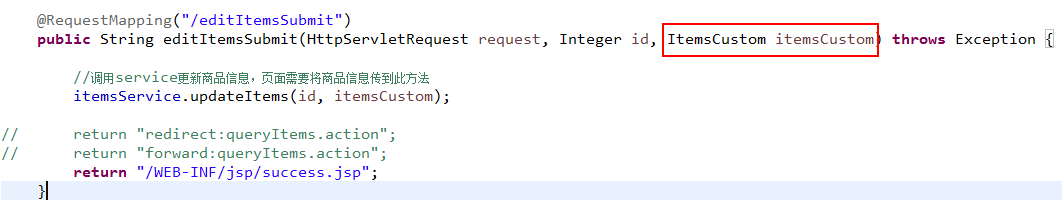
所以说，如果不使用@RequestParam，要求request传入参数名称和controller方法的形参名称一致，方可绑定成功。如果使用@RequestParam，不用限制request传入参数名称和controller方法的形参名称一致。通过@RequestParam中的required属性指定参数是否必须要传入，如果设置为true，没有传入参数就会报错。

## ****4. pojo类型的绑定****

### ****4.1 普通pojo类型****

　　再来总结下pojo类型的绑定，继续上面的案例，当页面展示了商品详细信息后，我做了修改，然后点击提交，后台应该将我提交的这些参数全部更新到数据库的items表中，也就是说，我提交的这些参数要绑定到Items对象或者其扩展对象中。先看一下Items中都有哪些属性：

‘

可以看到，有各种类型的属性，当我们提交后，要将这些属性全部封装到一个pojo中，然后去更新数据库。   
　　绑定很简单，对于基本类型，要求页面中input标签的name属性值和controller的pojo形参中的属性名称一致，即可将页面中数据绑定到pojo。也就是说前台页面传进来的name要和要封装的pojo属性名一模一样，然后就可以将该pojo作为形参放到controller的方法中，如下：   
  
　　这样就能将前台表单传进来的不同属性值封装到ItemsCustom中了。但是运行一下就会发现报错，报错的信息是无法将String类型转换成java.util.Date类型，因为上面Items中有一个属性是Date类型的createtime。这就需要我们自己定义转换器了，这也是这部分的重点，下面我们自己定义一个日期转换器：

//需要实现Converter接口，这里是将String类型转换成Date类型

public class CustomDateConverter implements Converter<String, Date> {

@Override

public Date convert(String source) {

//实现 将日期串转成日期类型(格式是yyyy-MM-dd HH:mm:ss)

SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

try {

//转成直接返回

return simpleDateFormat.parse(source);

} catch (ParseException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

//如果参数绑定失败返回null

return null;

}

}

### 定义好了转换器后，需要在springmvc.xml中进行如下配置：  转换器  转换器配置 　　现在就可以了，springmvc就能根据这个转换器将String类型正确转换成Date类型，然后封装到ItemsCustom中去了。  　　这里说一个小小的插曲：修改商品详细信息后提交，可能会有中文乱码问题，表达提交都是post方式，springmvc中关于post方式的中文乱码问题可以在web.xml中配置一个过滤器来解决，如下：  post乱码解决 ****4.2. 包装的pojo类型****

　　这个包装类型pojo与上面普通的pojo有啥区别呢？包装类型pojo指的是pojo中有另一个也是pojo的属性，即pojo套pojo，为什么会设计这种pojo呢？在前面的博文中我也有提到，这种组合的设计方法对于后期程序的扩展很有用，比如复杂的查询条件就需要包装到这种包装类型中。   
　　那么该如何绑定呢？有两个思路：

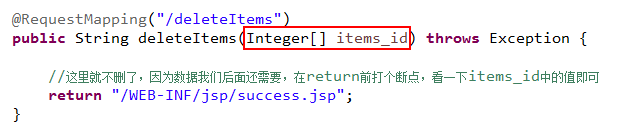
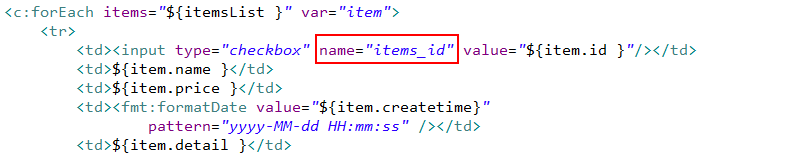
1. 在形参中添加HttpServletRequest request参数，通过request接收查询条件参数。
2. 在形参中让包装类型的pojo接收查询条件参数。

　　第一种方式就跟原始servlet差不多，这里使用第二种方法，我们传进来一个包装类型的pojo。看一下这个包装类型的pojo：   
   
　　这个包装pojo中还有一个ItemsCustom类，这个类继承了Items类，并且用来扩展与Items相关的User对象中的相关信息。所以这个ItemsCustom中有name属性，假如我们要想将前台传进来的name属性封装到ItemsCustom中的name属性中，该如何传入呢？这就是包装类型的pojo参数绑定问题。   
　　很简单，在前台我们可以通过这种方式来传：   
包装pojo参数   
　　然后controller中方法的形参传入包装类型的pojo，即ItemsQueryVo，打个断点，即可查看值有没有传进来。如下：   
   
　　这样就能根据用户传进来的参数，进行复制的查询操作了。

## ****5. 集合类型的绑定****

### ****5.1 数组的绑定****

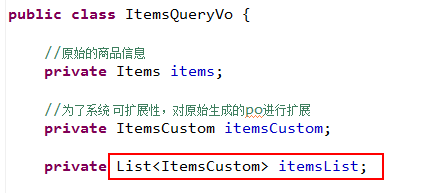
　　数组的绑定指的是前台传来多个同一类型的数据，我们在controller中使用数组形参来接收前台传来的数据。还是以案例来驱动这部分内容，比如现在我们要批量删除商品，那么我们需要勾选好几个商品，这些商品都有id号，在controller中，我们需要将这些id号全部获取并放到一个数组中，然后再根据数组中的id号挨个删除数据库中对应的项。那么该如何绑定呢？其实也很简单，如下：   
　　controller的方法定义为：

样就能将前台传入的多个id绑定到数组中，然后我们就可以从数组中拿出要删除的商品的id了。

### ****5.2 List的绑定****

　　通常在需要批量提交数据时，将提交的数据绑定到list<pojo>中，比如：成绩录入（录入多门课成绩，批量提交），在这里我们假设有需求：批量商品修改，在页面输入多个商品信息，将多个商品信息提交到controller方法中，即一次性更新多个商品信息。   
　　所以思路是在扩展类ItemsQueryVo中新添加一个List<ItemsCustom>，然后将不同商品的信息都存到这个List中，所以修改如下：



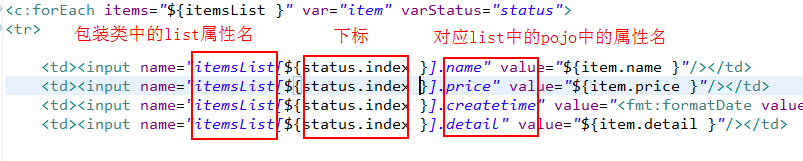
1、进入批量商品修改页面   
2、批量修改商品提交

所以controller中应该有两个方法，如下：





前台jsp页面中是如何传入参数的呢？这是我们所关心的问题，因为后台形参中接收数据用的就是包装类ItemsQueryVo。看下面：



所以我们知道了，前台是通过类似于list[i].name这种形式来传递参数的。list[i]表示第i个ItemsCustom，然后 list[i].属性 表示第i个ItemsCustom中对应的属性。

### ****5.2 Map的绑定****

　　Map的绑定其实和List的绑定是差不多的，首先也是在包装的pojo中新添加一个Map类型的属性，如（我随便举个例子，跟本例无关了）

Public class QueryVo {

private Map<String, Student> itemInfo = new HashMap<String, Student>();

//get/set方法..

}

关键是前台传参的时候和List不太一样，Map是这样传的，比如：

<tr>

<td>学生信息：</td>

<td>

姓名：<input type="text" name="itemInfo['name']"/>

年龄：<input type="text" name="itemInfo['price']"/>

.. .. ..

</td>

</tr>

　　我们可以看到，Map的参数绑定传来的是Map中的key，然后value会自动绑定到Map中的那个对象的属性中。在controller中的方法里，形参就直接使用QueryVo接收即可，也很简单。   
　　关于springmvc的参数绑定基本就总结到这了，其实原理都差不多，只是针对于不同的类型，绑定的方式有些区别而已，多想想多写写，基本就能掌握这些了。