# Day18\_SpringMVC

## SpringMVC的传值

### 1.创建用户编辑页面



### 2.创建一个控制器类,定义请求方法

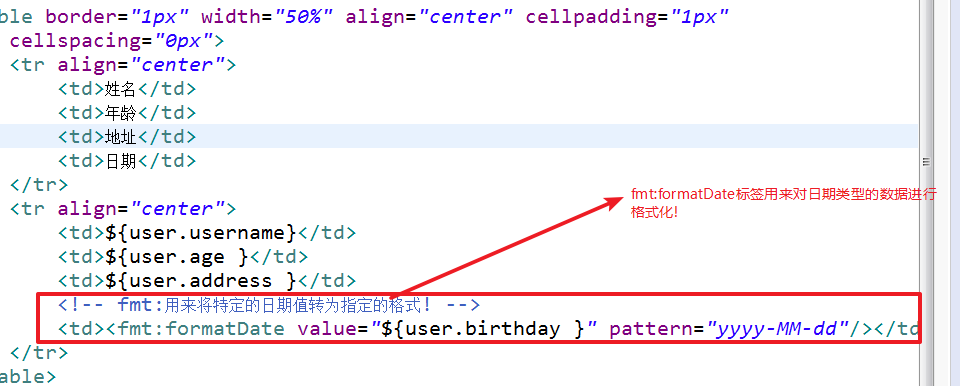


### 3.对特殊的数据类型进行转换



在该方法中,利用binder参数,将字符串转换为util.Date类型.

### 4.展示信息页面



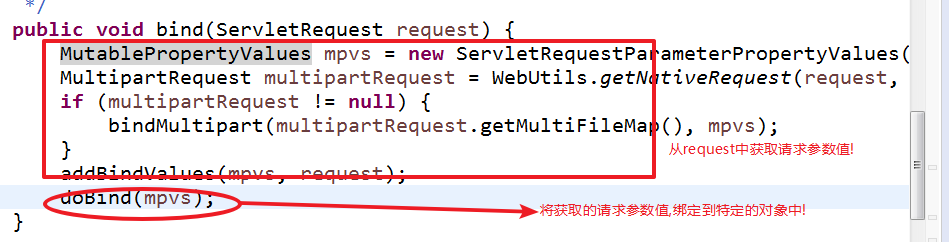
### 注意:

默认情况下,日期字符串是不能自动转换为util.Date和sql.Date的,所以会出现如下异常!



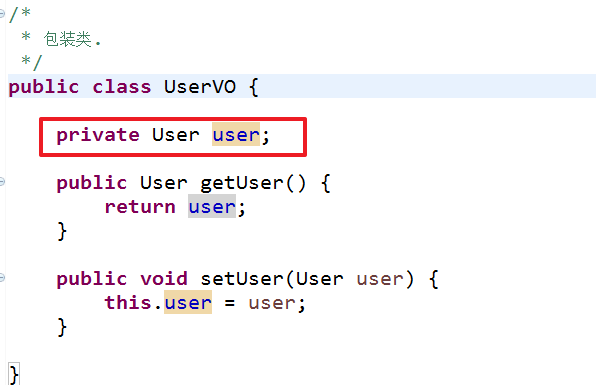


### 5.SpringMVC是如何将表单中的数据,自动封装进对应的对象中的原理

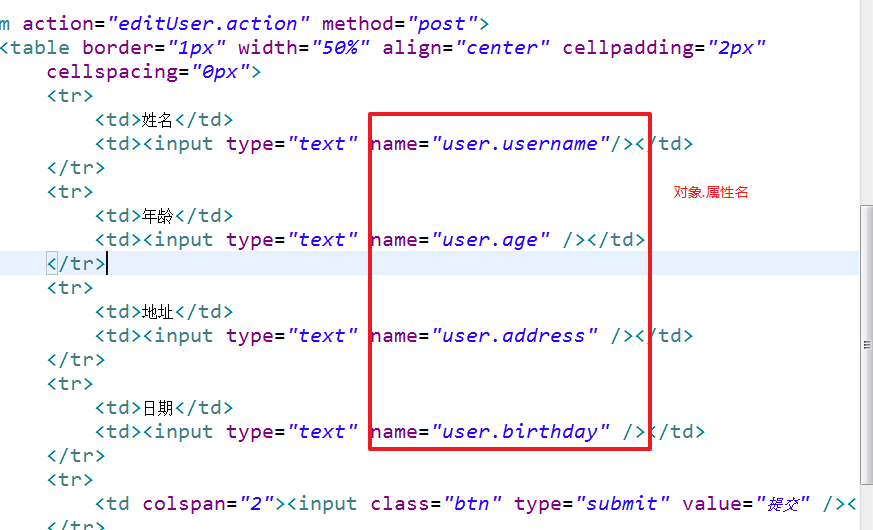


## 当存在包装类的情况下,表单数据的传递

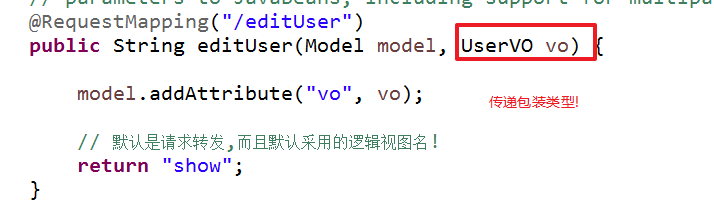
### 1.创建包装类



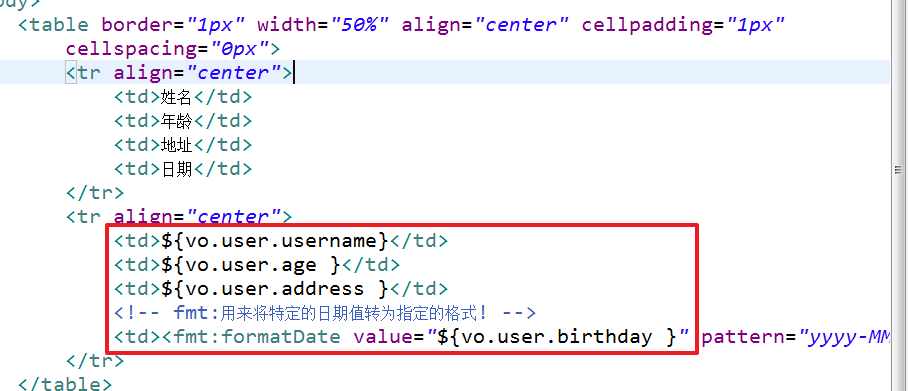
### 2.改造表单页面



### 3.传递包装类对象

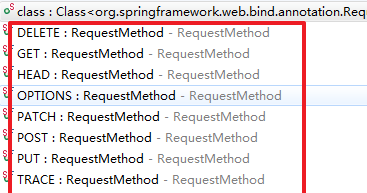


### 4.展示包装类中的对象信息

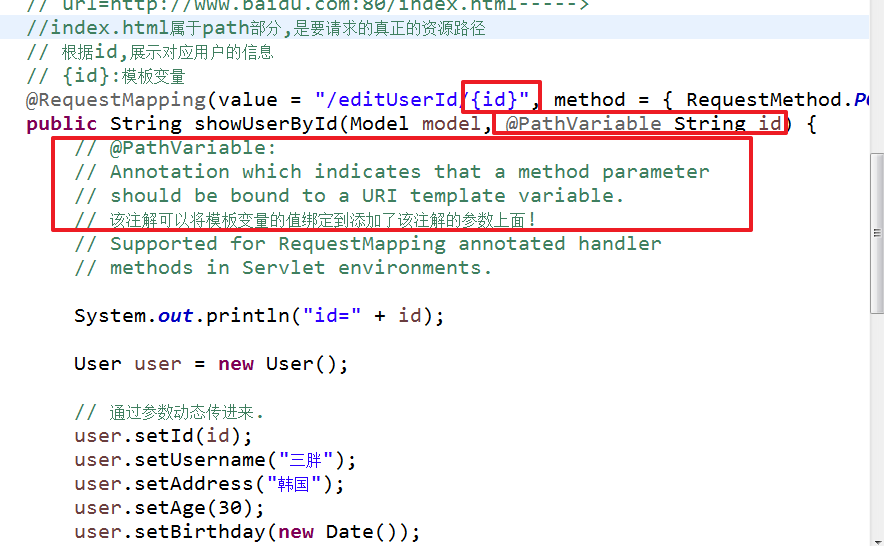


## restful风格的url

### 1.Java中的请求方法



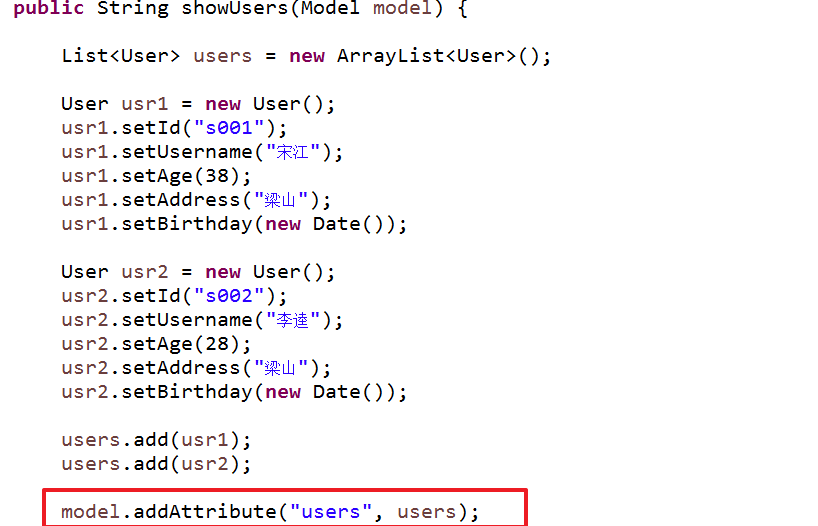
### 2.restful风格的url定义



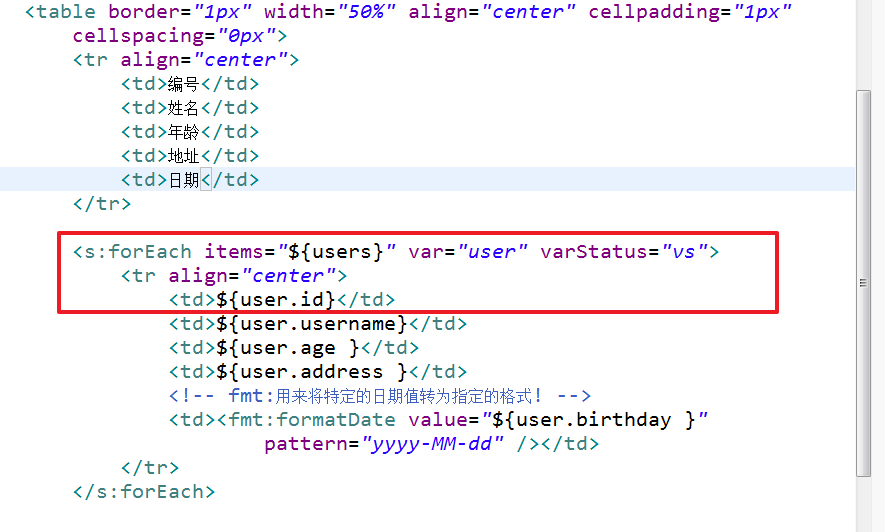


## 传递集合,展示集合信息

### .传递集合

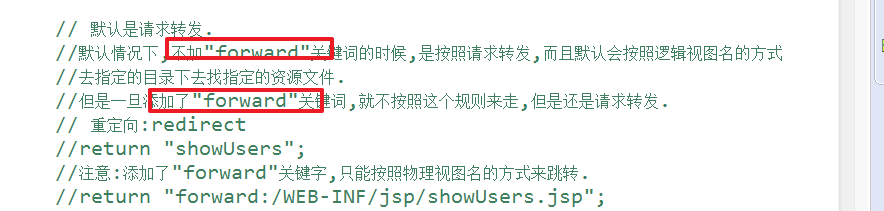


展示数据;



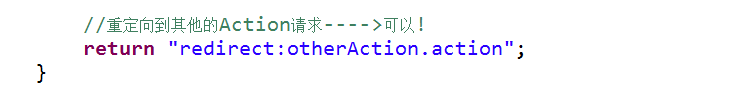
## 五.请求转发与重定向的实现

### 1.请求转发forward





### 2.重定向redirect



## 六.json数据的创建

### 1.xml和json的区别?

#### ①.文件大小:

xml文件会有很多额外的冗余的标签和其他的符号,会占用比较大的空间,网络传输速度较慢;

json没有太多的标签和符号,占用空间较小,网络传输速度较快.

#### ②.解析方面:

xml解析比较麻烦,解析方式复杂,一般需要专门的学习;

json解析相对简单很多.解析速度快的多.

### 2.SpringMVC创建JSON数据

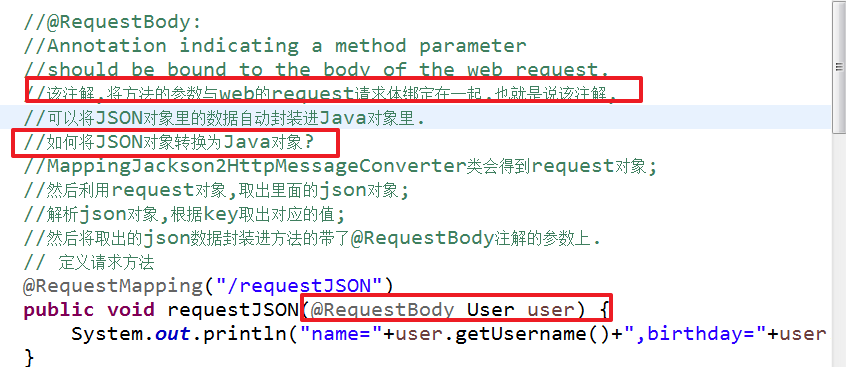
Java对象与JSON对象转换;

Java的集合与JSON数组转换.

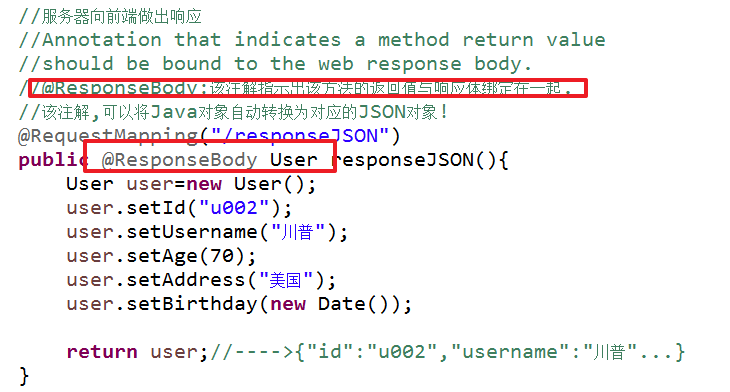
添加依赖包



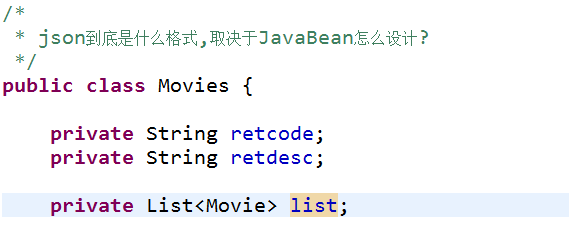
### [3.@RequestBody](mailto:3.@RequestBody)注解



### [4.@ResponseBody](mailto:4.@ResponseBody)注解



### 5.创建复杂的json数据





## 作业:

生成一个如下格式的json数据

<http://piao.163.com/m/cinema/list.html?app_id=2&mobileType=iPhone&ver=2.5&channel=lede&deviceId=C1985DD9-0125-4AB5-B66B-B91A85824BBA&apiVer=11&city=110000&dest_city=110000&latitude=40.04149&longitude=116.35745&offen_cinema_ids=1006%3A3037&page_no=1&page_num=0>

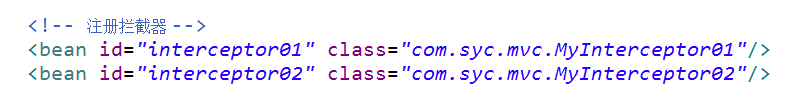
## 七.SpringMVC的拦截器

### 1.定义拦截器

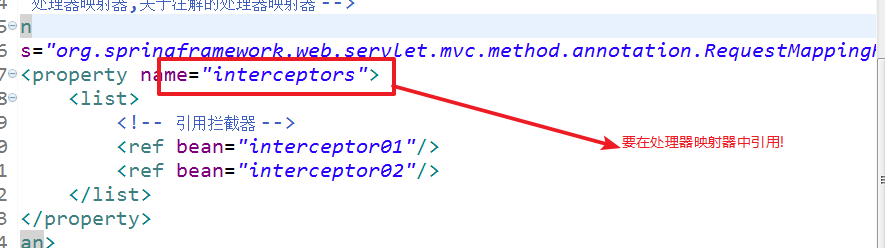




### 2.注册拦截器

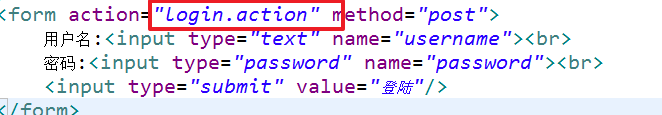


### 3.引用拦截器

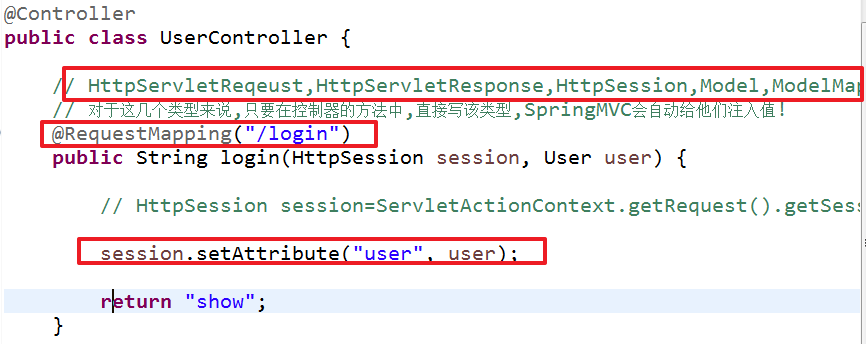


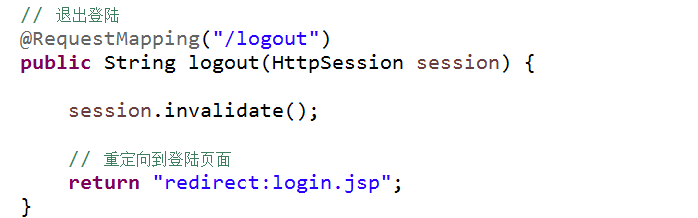
## 八.利用拦截器实现用户权限验证

### 1.创建登陆页面



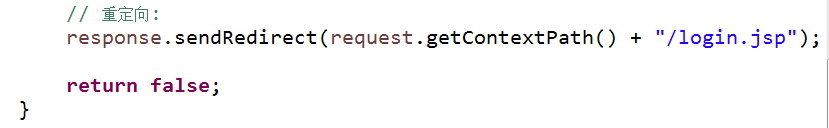
### 2.创建处理器,处理请求



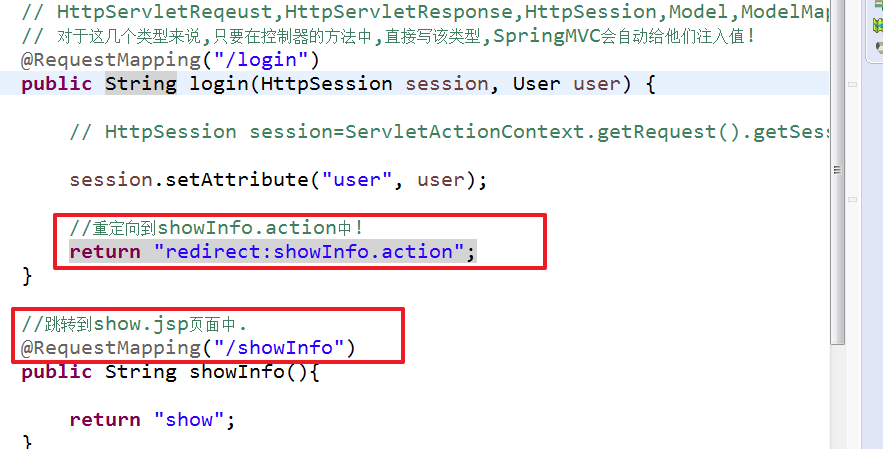


### 3.创建一个拦截器





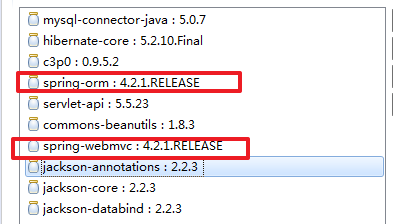
### 4.对控制器做出改造



## 九.SSH整合

### SpringMVC+Spring+Hibernate的整合.

### 1.添加依赖包



### 2.配置web.xml文件



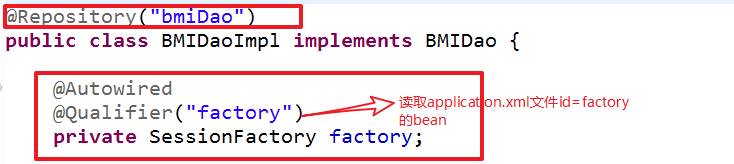
### 3.配置application.xml文件



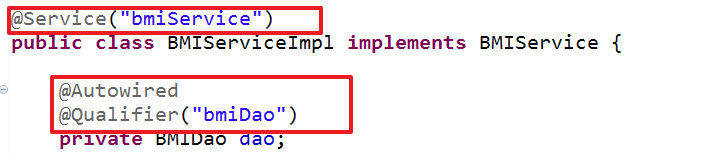


### 4.改造代码

#### dao层类的改造:



#### service层类的改造:



#### web层的类改造:

