SpringMVC简介：主要通过一系列的注解，无须实现任何借口，且可支持rest风格的url

## 第一节：helloworld

Idea中创建springmvc 步骤

1：选择创建一个project,勾选spring 右边勾选springmvc和webapplication命名等待完成

（加入jar包（context,expression,beans,core,webmvc mvc，aop）六个核心jar包 +commons-logs）

2：配置web.xml文件 url-pattern设置为拦截的url

第一步：配置dispatcher-servlet （扫描包（context-scanner:对应controller的包名）可配置视图解析器

第二步：配置初始化参数init-param修改param-value:classpath:springmvc.xml如果不写默认为servlet的名字

第三步：编写handler处理器（创建被扫描的包加类，在类上方加入注解controller,编写方法在方法上方加入注解requestmapping(“前端的url”)）

3：run edit configration设置server信息 ；然后在projectconstructer设置liberatory加入springmvc 和spring

4：编写index.jsp和success.jsp

## 第二节：requestmapping注解

可以修饰方法和类，类是相对于根目录路径，方法则是细分路径

## 第三节：requestmapping请求方法

Value指定url method指定请求方式

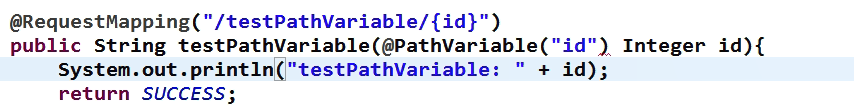
## 第四节：requestmapping请求参数和请求头

Value指定url,params={键值对} headers={键值对}

## 第五节：ant路径

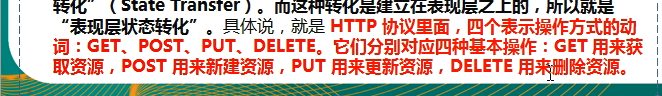
通配符（\*任意字符？一个字符，\*\*多层）

## 第六节：pathvariable



可以映射url的占位符到目标方法的参数中

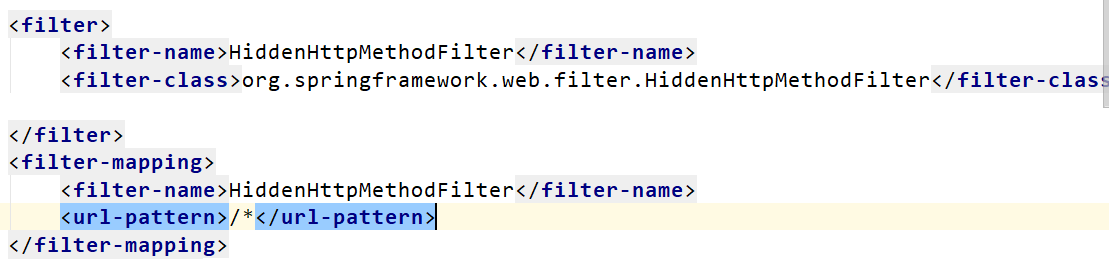
## 第七节：rest表现层状态转化



Post转换成delete或put请求



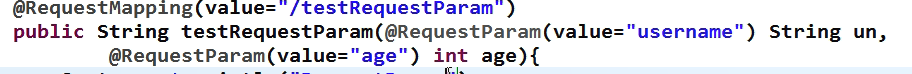
在web.xml中配置filter





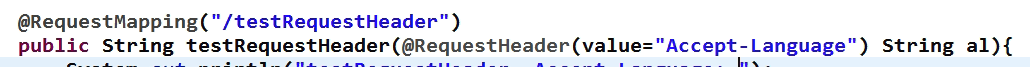
在springmvc通过@pathvariable注解获取到id

## 第八节：requestparam获取请求参数



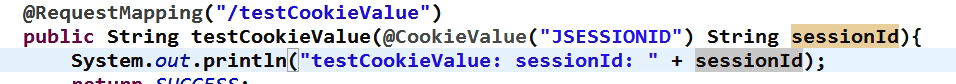
没带该参数会报错400；故需要设置 required=false(参数是否必须，默认必须)，defaultvalue请求参数的默认值

## 第九节：requestheader 请求头注解



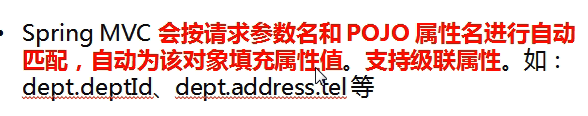
用法同requestparam 获取请求头信息

## 第十节：cookievalue 获取cookie

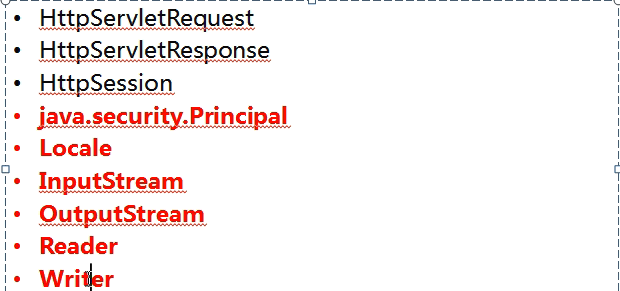


需要填写cookiename,属性同requestparam

## 第十一节：pojo（对象）作为参数



## 第十二节：servlet原生api

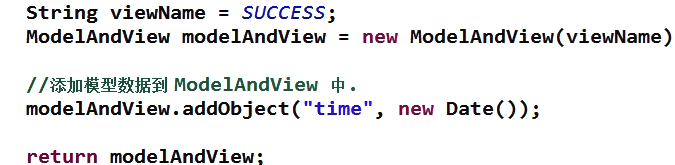


可以使用servlet原生api支持以上类型

## 第十四节：处理模型数据之modelandview



Modelandview:即包含视图信息也包括数据模型



Modelandview的model中数据放到request作用域中

## 第十五节：处理模型数据之map

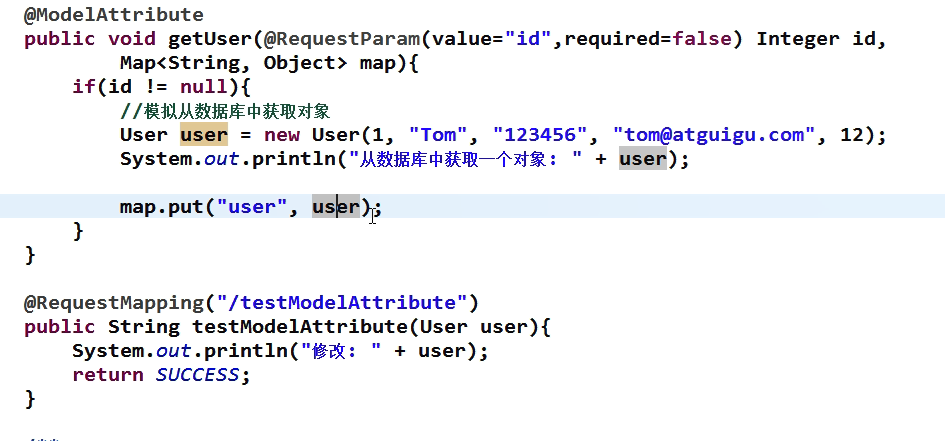
可以添加map类型实际上可以是model或者modelmap的参数

实际上是转成modelandview,将数据放在modelmap中的

## 第十六节：处理模型数据之sessionattributes

在requestmapping之上设置sessionattributes（value={键名}，type={类型}），可以指定对象类型把指定属性放置到会话中，此注解只能放置在类之上

第十七节：modelAttribute注解（用于把某个字段不能被修改，只修改其他字段的值）逻辑是先从数据库查询到对应的数据，然后修改指定字段的值



Modelattribute注解在执行目标方法之前被springmvc调用

## 第十七节：modelattribute的运行原理

第一步：执行mdelattribute注解修饰的方法，首先从数据库中取出对象后放置于map中

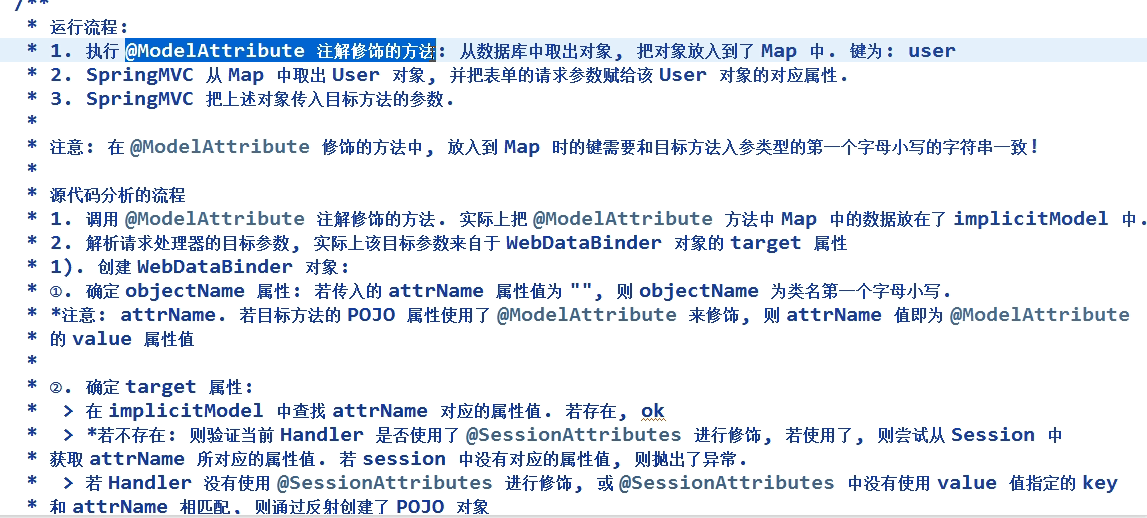
第二步：springmvc从map中取出对象，并把参数的值赋值为该对象的对应的属性

第三步：springmvc把上述对象传入目标方法的参数

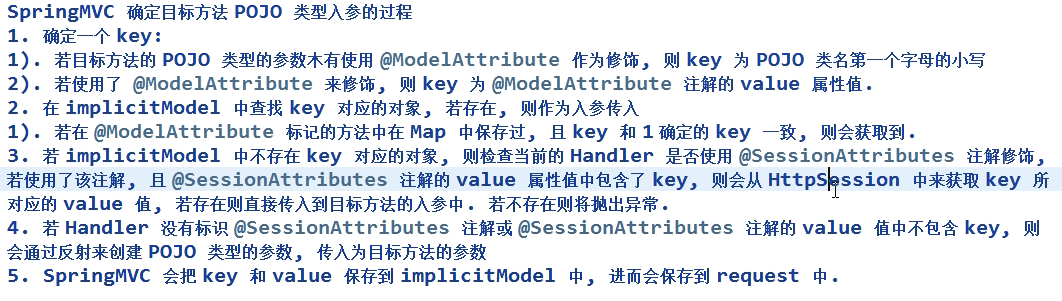
注意：在modelattribute放置到map时键名需要和目标方法入参类型的第一个字母小写的字符串一致



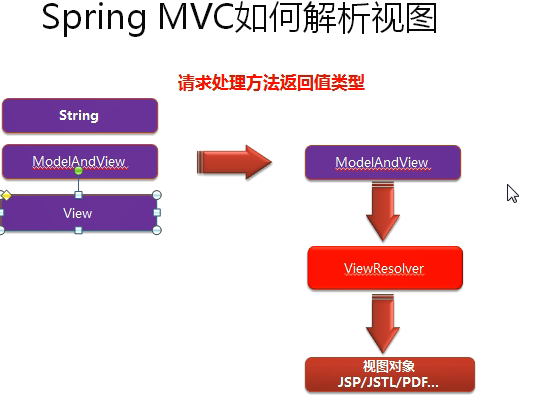
## 第十八节：modelattribute注解之源码分析



第十九节：确定目标方法pojo入参过程



## 第二十节：springmvc的视图流程



重要的view和viewresolver接口；springmvc都会装配成modelandview,通过视图解析器解析成真正的视图

## 第二十一节：JSTLview

资源国际化步骤

首先加入2个jar包（jstl+standand）,然后写三个properties文件（i18N.properties；i18N\_zh\_CN.properties;i18N\_en\_US.properties），在spring中配置resourceBundleMessageSource 配置基名属性(i18N)，页面使用fmt标签

## 第二十二节：mvc标签

直接响应转发页面，无需经过handler的方法，但之前配置的会报404，故需要配置annnotation-driven

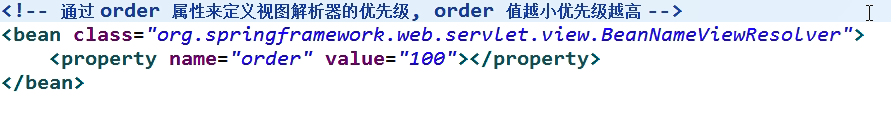




## 第二十三节：自定义视图

第一步：实现接口view并交由spring管理@compenent

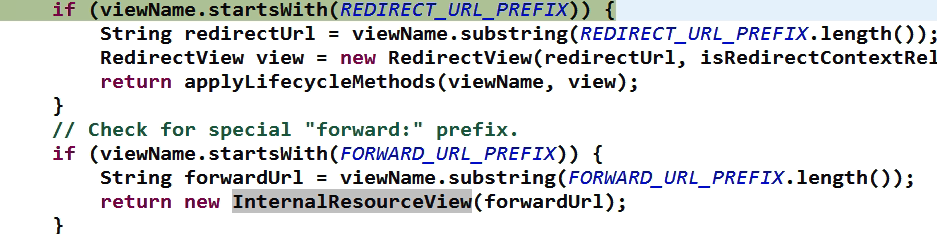
第二步：配置beannameviewresolver



## 第二十四节：请求的重定向

在return之后加入redirect:+字符串；默认为forward

内部原理：



第二十五节：restful风格的crud

第一步：创建一个动态web工程，加入jar

第二步：配置spring的dispatcher,并配置fitter将post请求转为delete或put hiddenhttpmethodfilter

第三步：配置spring.xml扫描包（component-scan）视图解析器internalresourceviewresolver

第四步：新建index.jsp,创建实体类entity(employee和company;两个dao

第五步：创建handler

注意在添加页面时需要回显部门信息，使用了form标签

