

1.

<url-pattern>写成/和/*的区别

大概说一下，/表示匹配/abc这样的路径，而/*表示匹配/abc,.jsp,.js,.html等等这样的路径。

2.

Spring classPath: 用法

classPath:定位相当于 /WEB-INF /classes/

3. web.xml基本包含内容

Spring核心监听器, 指定Spring配置文件applicationContext.xml

Spring MVC的前端控制器, 指定Spring MVC配置文件springmvc-config.xml, 拦截对象为/
编码过滤器, 一般指定为UTF-8, 拦截对象为/*

4. mybatis:scan的详解 (在applicationContext.xml中)

将base-package指定包下所有的接口当作mapper配置。

不要将非mapper接口放入指定的基本包中, 因为默认情况下mybatis:scan会将包里的所有接口当作mapper来扫描, 所以如果是其他接口就会出现错误。

可以在mybatis:scan元素的基本包里添加多个包, 使用逗号或分号分割即可。

5.

<context:component-scan>使用说明

在xml配置了这个标签后,

spring

可以自动去扫描base-pack下面或者子包下面的Java文件, 如果扫描到有@Component @Controller@Service等这些注解的类, 则把这些类注册为bean。

6. context:property-override的详解

<context:property-override location="..." /> //这样会自动地用properties文件中定义的属性覆盖spring配置文件中同名的属性值。

7. @RequestMapping

value属性: 标明映射路径

写法: value= "/..."

若只有该属性也可直接写为 "/..."

method属性: 标明方法处理哪些HTTP请求方式

写法: method=RequestMethod.POST

consumes属性: 标明处理请求的提交内容类型

写法: consumes="application/json"

表示该方法仅仅处

理 "application/json" 的请求

produces属性: 标明返回的内容类型

写法: produces="application/json"

表示该方法仅仅处

理 "application/json" 的请求,

返回的内容类型为

application/json

(比consumes更强制?)

params属性: 指定request中必须包含某些参数值时才让该方法处理

写法: params="myParam=myValue"

表示该方法仅仅处理名为 "myParam",

值为 "myValue" 的请求

headers属性: 指定request中必须包含某些指定的header值, 才让该方法处理

写法: headers="Referer=<http://www.jlb.com>"

表示该方法仅仅处理request的header

中包含了指定 "Referer" 请求头

和对应值为“<http://www.ilb.com>”的

请求

8. 一个控制器类中，@ModelAttribute修饰的方法总是会先于映射方法执行

9. Model, ModelMap, ModelAndView的区别

对于传入为Map型时，Model，ModelMap没什么区别，ModelMap只是指定了只存Map型，而ModelAndView除了存入Model，还要存入View

10. @RequestParam

在SpringMVC中，控制器层获取参数的方式主要有两种，一种是通过request.getParameter(“...”),另一种就是通过该注解实现

一. 基本使用，获取提交的参数

@RequestParam Object obj

spring会自动根据参数名字封装进入，我们可以直接拿这个参数名来用
(前端标明的name必须与obj一致)

二. 各种异常情况处理

1. 对传入参数指定参数名

@RequestParam(value=“sa”) Object obj
(前端标明的name必须与sa一致，obj随意起)

2. required可以要求前端参数是否一定要传

默认为true

3. 对2的说法也有特例，若参数是int型，并且required=false，

这时不传参也会报错，因为会赋值null给int

//个人理解，不要求前端传参但也应该设好默认值，配合defaultValue使用更佳

11. @PathVariable

可以非常方便地获得请求URL中的动态参数

写法: value=“/{id}”

@PathVariable Integer id

下面的定义的integer型的id直接获取地址中的id

12. @RequestHeader

用于将请求的头信息区的数据映射功能处理方法的参数上

写法: @RequestHeader(“User-Agent”) String userAgent

@RequestHeader(value=“Accept”) String[] accepts

自动将请求头“User-Agent”的值赋给userAgent，并将Accept请求头的值赋给accepts参数上

13. @CookieValue

写法: @CookieValue(value=“JSESSIONID”, defaultValue=“”)String sessionId

自动将JSESSIONID中的值设置到sessionId上

14. @SessionAttributes

允许我们有选择地指定Model中的哪些属性需要转存到HttpSession对象当中(级别为Session而已，前端获取时本来只能通过request，这个设定之后还可以通过session)

一般写法: 紧接着@Controller下一行@SessionAttributes(“user”), user就是指定的model属性名，这样设定以后若该控制器传输了一个名为user的model，前端就可以从session中获取。

15. @ModelAttribute(重中之重，几种用法好好区别)

该注解只支持一个属性value，类型为String，表示绑定的属性名称
慢慢来

一. 注释一个方法的参数

1. 从Form表单或URL参数中获取

@ModelAttribute User user

等价于@RequestParam(...)... 不过这个可绑定

默认值

(实际上，该处不注解依旧能获取，注意User类一定要有无参构造函数)

2. 从model中获取

有一个方法被注解 @ModelAttribute(“user”), 该方法会先于映射路径方法执行并返回一个User对象，映射路径方法的参数User user会被注解@ModelAttribute(“user”)与上一个方法对应，意指该user的属性就是上一个方法的返回值的属性

二. @ModelAttribute(value=“”)注释返回具体类的方法

