**Spring MVC Study Notes**

**@author Count Monte Cristo**

**@create D20150705**

**@version ~~D20150714~~ D20151216**

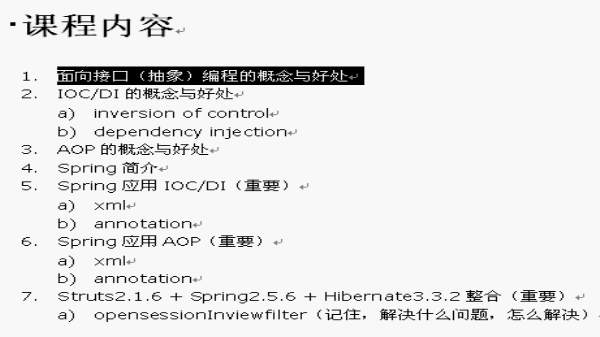
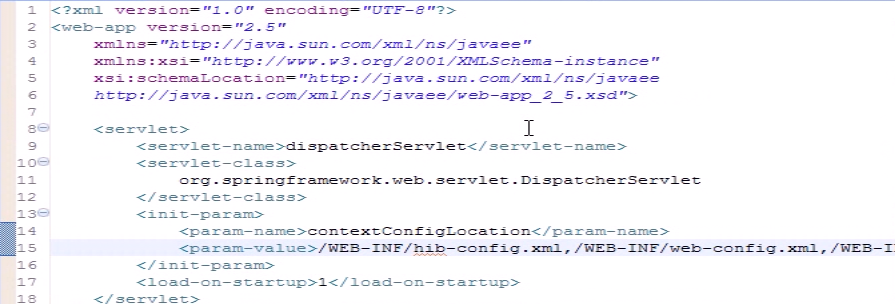
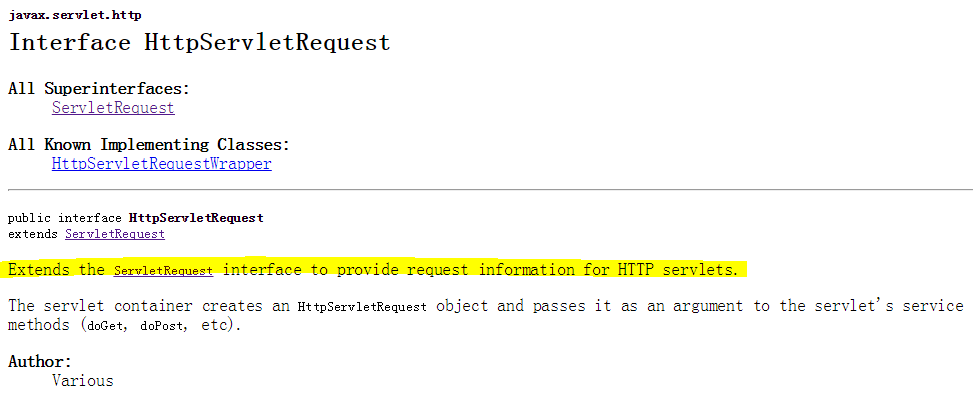
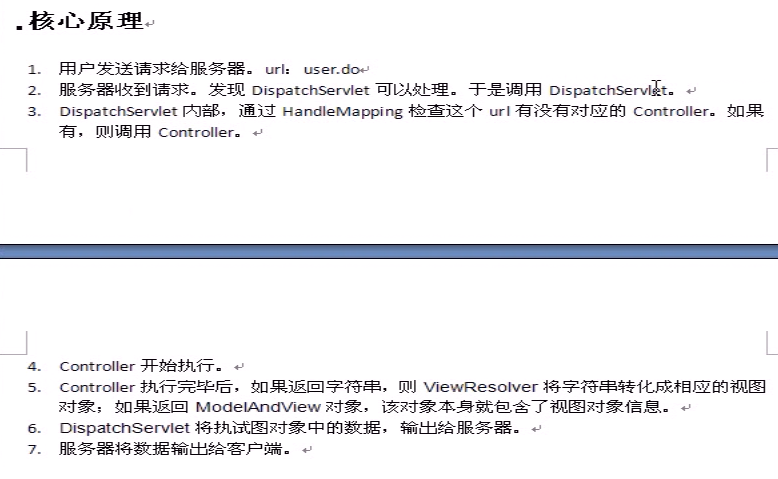
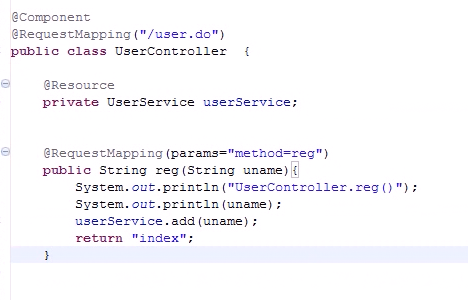
**@description 纪录《尚学堂》Spring MVC视频教学中重要的知识**

**@resource 《尚学堂》**

**@motto VICTORY LOVES PREPRATION**

**@criteria 凯旋基诺**

**目录**

1. **课程内容**
2. **分层思想简介**
3. **web.xml配置**
4. **HttpServletRequest和HttpServletResponse解析**
5. **核心原理**
6. **注解解析**
7. **注解(详细)解析**
8. **Controller类中方法参数的处理**
9. **基于Spring MVC框架的文件上传**
10. **处理AJAX请求**
11. **Spring MVC中的拦截器(两种基本方式)**
12. **课程内容**
    1. 视图
13. **分层思想**
    1. MVC（Model + View + Controller）
    2. 具体又可以分
       1. Domain
          1. 对象的域
       2. Service
          1. 对象的行为，具体的业务操作
       3. Dao
          1. 数据库操作
       4. View
          1. 视图
          2. 可以有JSP、VM等模板引擎
       5. Controller
          1. 控制器，主要是控制逻辑，没有具体的业务实现
       6. 说明
          1. Domain + Service + Dao = Model，面向对象编程思想中实际世界的对象就是Model，Model中包含域（domain）和行为【Service + （Dao，针对数据库的行为）】
14. **web.xml配置**
    1. **示例**
    2. **解析**
       1. dispatcherServlet是SpringMVC的调度Servlet
       2. <init-param>中配置的是加载对应Servlet的时候加载的配置文件
       3. <load-on-startup>中的参数越小越先加载(最小为0)
       4. contextConfigLocation参数用于配置
15. **HttpServletRequest和HttpServletResponse解析**
    1. **HttpServletRequest**
       1. **官方定义**
       2. **说明**
          1. request是服务器端容器生成的一个用于保存客户端传递的参数以及其他的客户端信息
          2. 注意：Controller到JSP页面之间传递参数是属于服务端内部传递参数，使用的是request而不是response
          3. request.getRequestDispatcher(“view”).forward(request,response)是将客户端的请求forward到view对应的页面或者链接，进一步进行处理
    2. **HttpServletResponse**
       1. **官方定义**
       2. **说明**
          1. response是服务器端容器生成的用于给反馈给客户端相关的页面和参数信息
          2. 注意：客户端可以是移动客户端（Android、IOS、WP等）
16. **Spring MVC 核心原理**
    1. **步骤**
    2. 注意：需要通过看源码进一步了解该执行过程细节
17. **Spring MVC注解解析**
    1. **示例**
    2. **(注解)解析**
       1. @Component、@Controller、@Service、@Repository、@Autowired@Resource、@RequestMapping
       2. @Component：标注该类进入Spring的注解范围
       3. @Controller：标注该类是控制器类，此处可以使用@Component
       4. @Service：标注该类是服务类，此处可以使用@Component
       5. @Repository：标注该类是Dao类，此处可以使用@Component
       6. @Resource：标注变量是已经入注解范围内的资源
       7. @Autowired：标注该字段自动装配值，不用new，此处可以使用@Resource
       8. @RequestMapping(value = “/user” params = “method=reg”)
          1. value = “/user”标注请求路径
          2. params = “method=reg”标注请求参数中参数method的值为reg，也就是说只要请求参数中有参数为method且其值为reg就是映射到其修饰的方法上面
    3. **传入参数解析**
       1. **示例**

****

* + 1. **解析**
       1. params = “method=reg”可以接收前端名为method参数
       2. uname可以接收前端名为uname的参数
       3. 若Entity Bean:User中有uname属性，也可以用User user作为reg方法的参数接收客户端的参数字段uname
       4. 注：以上接收方式不需要再写其他辅助，Spring会自动注入（详情有待参考Spring源码）

1. **Spring MVC注解（详细）解析**
   1. **@Controller**
      1. Spring的Controller是Singleton的，这就意味着会被多个请求线程共享。因此，我们将控制器设计成无状态类。在spring 3.0中，通过@Controller标注即可将class定义为一个controller类。为使spring能找到定义为controller的bean,需要在spring-context配置文件中增加如下定义：

<context:component-scan base-package="com.sxt.web"/>

注：实际上，使用@component，也可以起到@Controller同样的作用。

* 1. **@RequestMapping**
     1. 定义在类或者方法前面
     2. 指定类或者方法的访问路径
  2. **@RequestParam**
     1. 一般用来将指定的参数赋值给方法中的形参，此时形参客户端传入的参数名称不一致（如果一致的话就不需要@RequestParam，可以直接注入值）
     2. 示例：

public String method(@RequestParam(“id”) int uid)

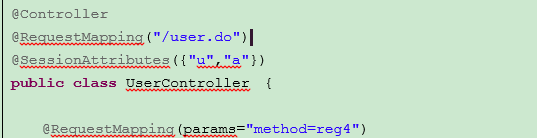
作用：把前端参数id赋值给uid

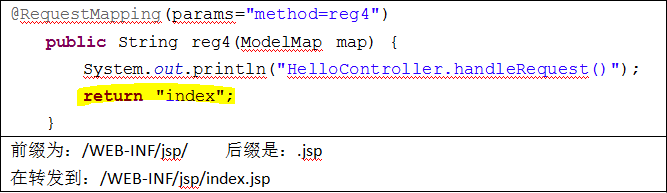
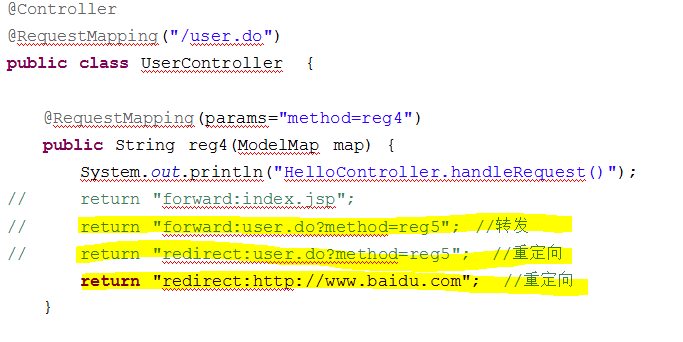
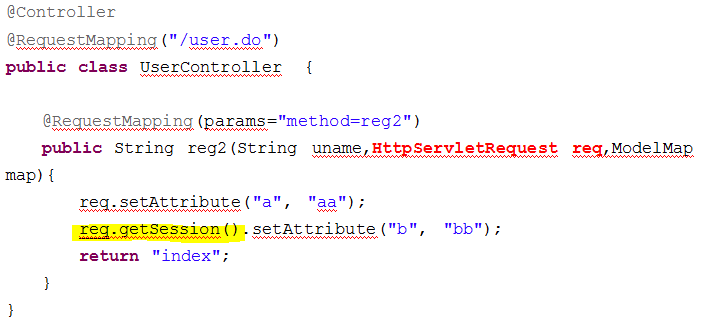
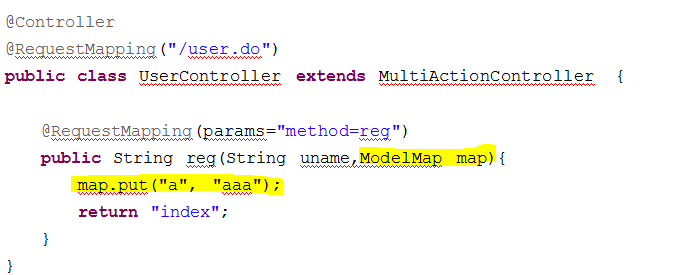
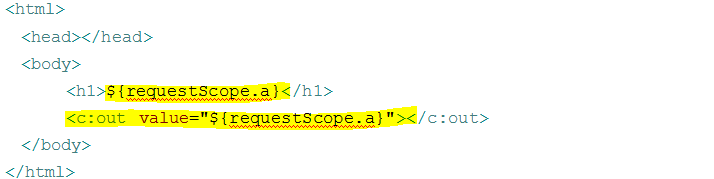
* 1. **@SessionAttributes**
     1. 将ModelMap中指定的属性放到Session中
     2. **示例：**

说明：ModelMap中的key-value为u-uuuu，同时把key为u的value放在了session中，因为@SessionAttributes(“u”, “a”)中进行配置说明了

**@ModelAttribute**

* + 1. 可以和@SessionAttributes配合使用，可以将ModelMap中的值通过该注解自动赋值到指定变量
    2. 示例：



1. **Controller类中方法参数的处理**
   1. **Controller类中方法返回值的处理**
      1. **返回String类型**（建议返回类型）
         1. **实例：**
      2. **返回ModelMap、ModelAndView、Map、List、Set、Object、无返回值（不建议使用）**
   2. **请求转发和重定向**
      1. **实例**
      2. **转发**
         1. 代码
            1. return “forward:user.do?method=reg5”
      3. **重定向**
         1. 代码
            1. return “redirect:user.do?method=reg5”
            2. return “redirect:https://www.b、aidu.com”
   3. **获取HttpServletRequest和HttpSession对象（这两个类见他们的具体分析Notes）**
      1. **实例**
      2. **通过HttpServletRequest对象获取HttpSession**
   4. **ModelMap**
      1. 是java.util.Map的实现，可以在其中存放属性，作用域同request。下面这个示例，我们可以在modelMap中放入数据，然后在forward的页面上显示这些数据。通过el表达式、JSTL、java代码均可。
      2. **实例**
      3. **实例说明**
         1. 后端可以通过Spring的ModelMap（作用同HttpServletRequest）（和Model一样）进行赋值
         2. 前端可以通过EL表达式、JSTL或者Java代码等提取HttpServletRequest中的值
            1. EL提取

${requestScope.a}

* + - * 1. JSTL提取

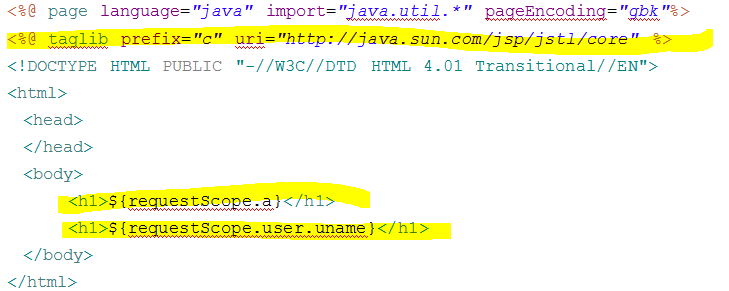
<c:out value=”${requestScope.a}”>

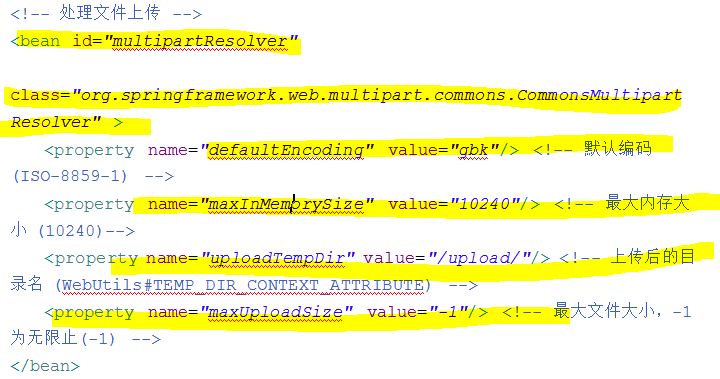
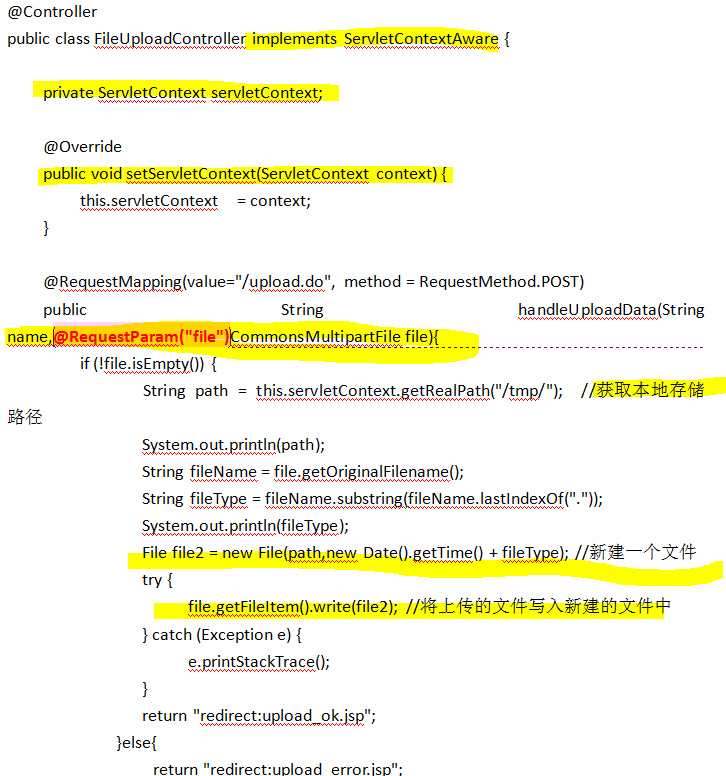
日期输出

<fmt:formatDate value="${login.createDate}" pattern="yyyy-MM-dd HH:mm:ss" />

* + - * 1. Java代码提取

<%= request.getParameter(“a”) %>

* 1. **ModelAndView模型视图类**
     1. ModelAndView中的Model代表模型，View代表视图。即，这个类把要显示的数据存储到了Model属性中，要跳转的视图信息存储到了view属性。
     2. 实例
        1. **后端**
        2. **前端**
     3. **实例说明**
        1. 见实例中的注释部分

1. **基于Spring MVC框架的文件上传**
   1. **必须引入两个jar包(具体作用有待剖析)**
      1. apache-commons-fileupload.jar
      2. apache-commons-io.jar
   2. **在springmvc-servlet.xml配置文件中，增CommonsMultipartResoler的配置：**
   3. **form表单的enctype(规定在将表单数据发送到服务器之前如何对其进行编码)必须设置为"multipart/form-data"，即enctype=*"multipart/form-data"***
   4. **控制器(内容较简单在此不进行解析)**
2. **处理AJAX请求**
3. **Spring MVC中的 (两种基本方式)**
   1. **实现接口**

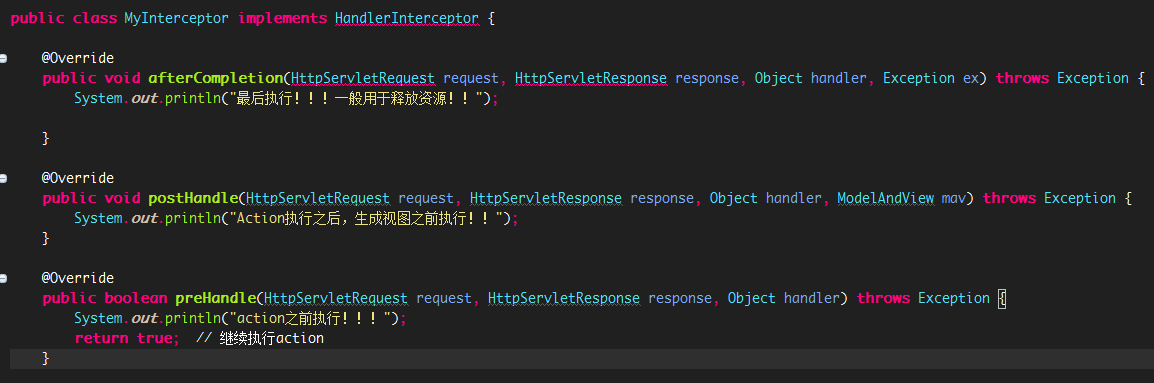
org.springframework.web.servlet.HandlerInterceptor

需要实现下面的方法：

* 1. **继承适配器(类)**

org.springframework.web.servlet.handler.HandlerInterceptorAdapter

* 1. **XML中配置**
  2. **实例**
     1. **实现HandlerInterceptor接口**

****

* + 1. **继承HandlerInterceptorAdapter适配器**

****

* 1. **注意**
     1. 当设置多个拦截器时，先按顺序调用prehandle（请求还没有到controller）方法，然后分别逆序调用每个拦截器的posthandle（访问完controller还没有访问view）和aftercompletion（全部访问完，常用于资源消除）方法

Spring MVC Study Notes

@author Count Monte Cristo

@create D20151103

@version D20151103

@description 记录Spring MVC学习内容

@resource 《传智博客》

@notice 有待和上面的note进行整理

@motto VICTORY LOVES PREPARATION

@criteria 凯旋基诺