springmvc的基础知识

springmvc课程安排：

第一天： 基础知识

springmvc框架(重点)

mvc在b/s系统中应用方式

springmvc框架原理（DispatcherServlet前端控制器、处理器映射器、处理器适配器、视图解析器）

springmvc入门程序

需求：商品列表查询

常用的处理器映射器，处理器适配器

注解的处理器映射器，处理器适配器用于注解开发(重点)

3

注解开发基础

常用的注解使用

参数绑定

自定义参数绑定

springmvc和struts的区别

第二天： 高级知识

使用springmvc上传图片

集合参数的绑定

validation校验器

异常处理器使用（用于系统异常处理）

RESTful支持

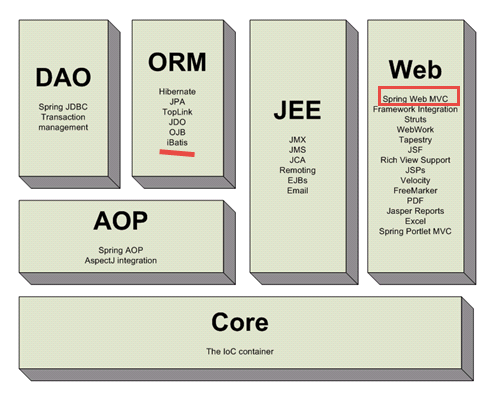
拦截器

学生练习

# springmvc框架

## springmvc是什么？

springmvc是spring的一个模块，提供web层解决方案（基于mvc设计架构）



## mvc在b/s系统的应用

mvc是一个设计模式，在b/s系统的应用：

request

**C （controller）控制器**

**作用：接收用户请求**

**及响应结果**

response

**M （model）**

**模型**

**pojo**

**action**

**service**

**dao**

请求模型进行处理

返回处理结果

**V (view)视图**

填充模型数据渲染视图

注意：b/s系统模型无法将数据推送到视图

## springmvc的框架

DispatcherServlet

前端控制器

1 request请求

url

Handler处理器

HandlerMapping处理器映射器

2请求查找Handler

3 HandlerExecutionChain{

HandlerInterceptor

HandlerInterceptor

Handler

}

HandlerAdapter处理器适配器

4请求执行Handler

5执行

6ModelAndView

7向前端控制器返回

ModelAndView

ViewResolver视图解析器

8请求进行视图解析

9返回view

View视图

10渲染视图

向view填充模型数据

11 response

第一步：用户发起request请求，请求至DispatcherServlet前端控制器

第二步：DispatcherServlet前端控制器请求HandlerMapping处理器映射器查找Handler

DispatcherServlet：前端控制器，相当于中央调度器，各各组件都和前端控制器进行交互，降低了各各组 件之间耦合度。

第三步：HandlerMapping处理器映射器，根据url及一些配置规则（xml配置、注解配置）查找Handler，将Handler返回给DispatcherServlet前端控制器

第四步：DispatcherServlet前端控制器调用适配器执行Handler，有了适配器通过适配器去扩展对不同Handler执行方式（比如：原始servlet开发，注解开发）

第五步：适配器执行Handler

Handler是后端控制器，当成模型。

第六步：Handler执行完成返回ModelAndView

ModelAndView：springmvc的一个对象，对Model和view进行封装。

第七步：适配器将ModelAndView返回给DispatcherServlet

第八步：DispatcherServlet调用视图解析器进行视图解析，解析后生成view

视图解析器根据逻辑视图名解析出真正的视图。

View：springmvc视图封装对象，提供了很多view，jsp、freemarker、pdf、excel。。。

第九步：ViewResolver视图解析器给前端控制器返回view

第十步：DispatcherServlet调用view的渲染视图的方法，将模型数据填充到request域 。

第十一步：DispatcherServlet向用户响应结果(jsp页面、json数据。。。。)

DispatcherServlet：前端控制器，由springmvc提供

HandlerMappting：处理器映射器，由springmvc提供

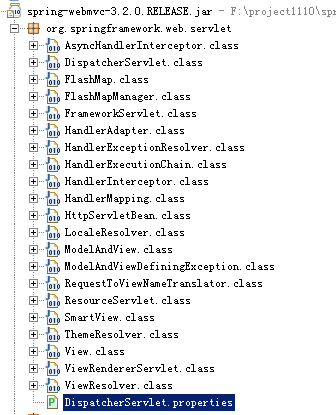
HandlerAdapter：处理器适配器，由springmvc提供

Handler：处理器，需要程序员开发

ViewResolver：视图解析器，由springmvc提供

View：真正视图页面需要由程序编写

# DispatcherServlet.properoties



DispatcherServlet前端控制器加载 DispatcherServlet.properoties 配置文件，从而默认加载各各组件，

如果在springmvc.xml中配置了处理器映射器和适配器，以sprintmvc.xml中配置的为准

# 注解映射器和适配器

## 注解映射器

spring3.1之前默认加载映射器是

org.springframework.web.servlet.mvc.annotation.DefaultAnnotationHandlerMapping，3.1之后要使用：

org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping

在springmvc.xml中配置RequestMappingHandlerMapping：

使用RequestMappingHandlerMapping需要在Handler 中使用@controller标识此类是一个控制器，使用@requestMapping指定Handler方法所对应的url。

## 注解适配器

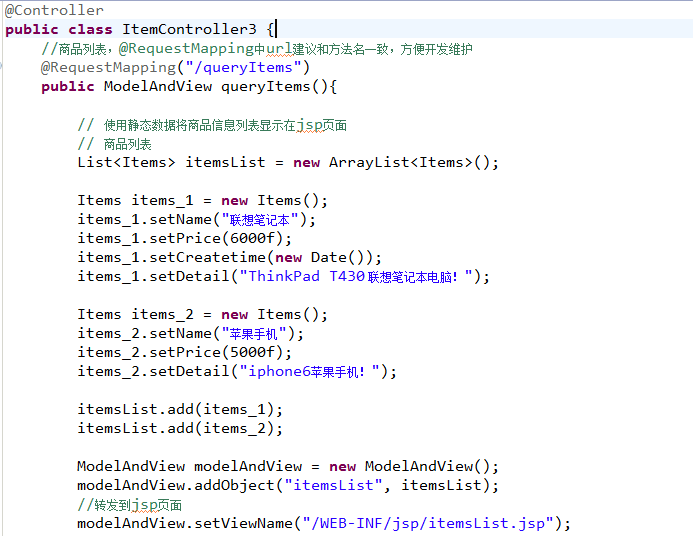
spring3.1之前默认加载映射器是

org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter，3.1之后要使用：

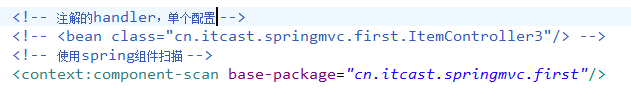
org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter

RequestMappingHandlerAdapter，不要求Handler实现任何接口，它需要和RequestMappingHandlerMapping注解映射器配对使用，主要解析Handler方法中的形参。

## 注解开发Hanlder



## 配置Handler



建议使用组件扫描，组件扫描可以扫描@Controller、@Service、@component、@Repsitory

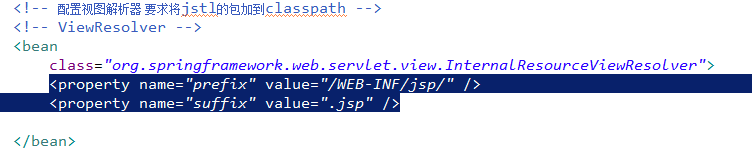
# 小结

DispatcherServlet：前端控制器，相当于中央调度器，可以降低组件之间的耦合度。

HandlerMapping：处理器映射器,负责根据url查找Handler

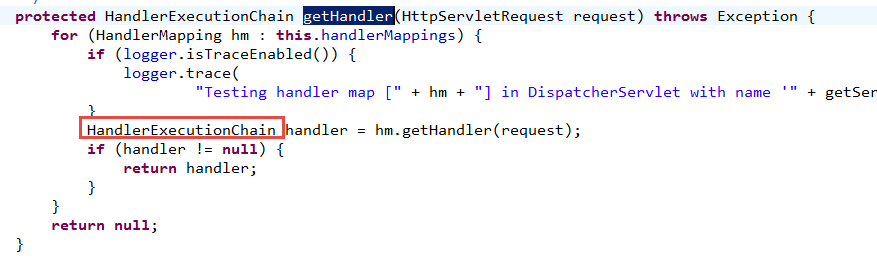
HandlerAdapter：处理器适配器，负责根据适配器要求的规则去执行处理器。可以通过扩展适配器支持不同类型的Handler。

viewResolver：视图解析器，根据逻辑视图名解析成真正的视图，



真正视图地址==前缀+逻辑视图名+后缀

1、DispatcherServlet通过HandlerMapping查找Handler

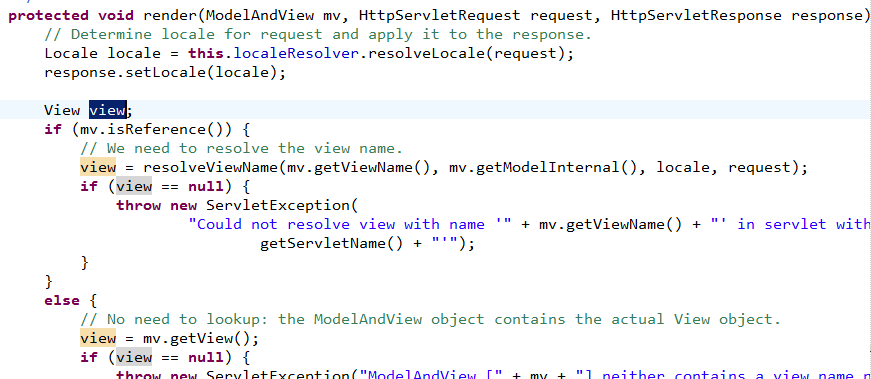


2、DispatcherServlet通过适配器去执行Handler，得到ModelAndview



3、视图解析

视图解析完成得到一个view：



4、进行视图渲染

将Model中的数据 填充到request域。

