对于第三阶段所学

大家有什么不懂的地方呀？

都不会！

问题汇总：

1. springmvc的执行流程

1、 首先用户发送请求——>DispatcherServlet，前端控制器收到请求后自己不进行处理，而是委托给其他的解析器进行处理，作为统一访问点，进行全局的流程控制；

2、 DispatcherServlet——>HandlerMapping， HandlerMapping将会把请求映射为HandlerExecutionChain对象（包含一个Handler处理器（页面控制器）对象、多个HandlerInterceptor拦截器）对象，通过这种策略模式，很容易添加新的映射策略；

3、 DispatcherServlet——>HandlerAdapter，HandlerAdapter将会把处理器包装为适配器，从而支持多种类型的处理器，即适配器设计模式的应用，从而很容易支持很多类型的处理器；

4、 HandlerAdapter——>处理器功能处理方法的调用，HandlerAdapter将会根据适配的结果调用真正的处理器的功能处理方法，完成功能处理；并返回一个ModelAndView对象（包含模型数据、逻辑视图名）；

5、 ModelAndView的逻辑视图名——> ViewResolver， ViewResolver将把逻辑视图名解析为具体的View，通过这种策略模式，很容易更换其他视图技术；

6、 View——>渲染，View会根据传进来的Model模型数据进行渲染，此处的Model实际是一个Map数据结构，因此很容易支持其他视图技术；

7、返回控制权给DispatcherServlet，由DispatcherServlet返回响应给用户，到此一个流程结束。

1. 常用注解

@Controller > spring

@RequestMapping

@ReponseBody == void 工具json 》》打包json

@PathVariable <http://localhost:8080/eastCMS/90>

@RequestParam 可以获取前端页面的指定的键值对

还可以接收多选值

<http://localhost:8080/eastCMS?name=xiaofang>

public void show(@requestParam(“name”) String ss){

}

public void show(@requestParam(“name2”) List<Integer> slist){

}

1. 配置常用不（注解）

常用！！！！

扫描 记住，会配置（controller）

视图解析器 （无条件记住）

静态资源放行 记住，会配置

上传下载解析器 （无条件记住）

<multipart-config></multipart-config>

（时间）格式转换器 （会，了解）

1. 哪些配置需要强行记住，没有为啥的
2. 需要粘贴复制的内容是哪些（ssm框架所有的）

！！！！！！！

1. 逆向工程是哪些配置

一个配置文件，一个main方法，注意不要重复运行

1. spring容器的概念

IOC(DI) 控制反转：

代码提现：

@Service @Repository @Component @Controller

@Autowired //@Resource

LoginUserMapper loginUserMapper;

AOP

<!-- 事务管理器 -->

<bean id=*"transactionManager"*

class=*"org.springframework.jdbc.datasource.DataSourceTransactionManager"*>

<!-- 数据源 -->

<property name=*"dataSource"* ref=*"dataSource"* />

</bean>

<!-- 通知 -->

<tx:advice id=*"txAdvice"* transaction-manager=*"transactionManager"*>

<tx:attributes>

<!-- 传播行为 -->

<tx:method name=*"save\*"* propagation=*"REQUIRED"* />

<tx:method name=*"insert\*"* propagation=*"REQUIRED"* />

<tx:method name=*"add\*"* propagation=*"REQUIRED"* />

<tx:method name=*"create\*"* propagation=*"REQUIRED"* />

<tx:method name=*"delete\*"* propagation=*"REQUIRED"* />

<tx:method name=*"update\*"* propagation=*"REQUIRED"* />

<tx:method name=*"find\*"* propagation=*"SUPPORTS"* read-only=*"true"* />

<tx:method name=*"select\*"* propagation=*"SUPPORTS"* read-only=*"true"* />

<tx:method name=*"get\*"* propagation=*"SUPPORTS"* read-only=*"true"* />

<tx:method name=*"\*"* propagation=*"SUPPORTS"* read-only=*"true"* />

</tx:attributes>

</tx:advice>

<!-- 切面 -->

<aop:config>

<aop:advisor advice-ref=*"txAdvice"*

pointcut=*"execution(\* com.offcn.service..\*.\*(..))"* />

</aop:config>

1. SpringAOP的概念

**AOP的工作重心在于如何将增强应用于目标对象的连接点上，这包括第一，如何通过切点和增强定位到连接点，第二，如何在增强中编写切面代码。**

1. spring里面需要记住的配置，木有为什么的那种

1)AOP配置

2）<!-- spring配置文件加载 -->

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:spring/applicationContext-\*.xml</param-value>

</context-param>

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

1. mybatis知识点有哪些，配置文件有哪些

personMapper save()

PersonMapper.xml save(List<Integer> slist)

映射文件：《resultMap》

他可以做一对一，一对多的操作

<where>

<if>

<sql>

<foreach>

1. 一对一，一对多用得多么（因为看似比较难）

背过！！写过！！！（第三种！！！！）

一对一：

<**resultMap id="map002" type="com.offcn.pojo.Person"**>  
 <**id property="pid" column="pid"**></**id**>  
 <**result property="pname" column="pname"**></**result**>  
 <**collection property="blist" column="pid" ofType="com.offcn.pojo.Book"  
 select="com.offcn.mapper.BookMapper.selectBoobById"**></**collection**>  
</**resultMap**>

一对多

<**resultMap id="map002" type="com.offcn.pojo.Person"**>  
 <**id property="pid" column="pid"**></**id**>  
 <**result property="pname" column="pname"**></**result**>  
 <**collection property="blist" column="pid" ofType="com.offcn.pojo.Book"  
 select="com.offcn.mapper.BookMapper.selectBoobById"**></**collection**>  
 </**resultMap**>

1. mybayis哪些地方是强行记住的

</bean> -->

<bean id=*"sqlSessionFactory"* class=*"org.mybatis.spring.SqlSessionFactoryBean"*>

<!-- 数据库连接池 -->

<property name=*"dataSource"* ref=*"dataSource"* />

<!-- 加载mybatis的全局配置文件 -->

<property name=*"configLocation"* value=*"classpath:mybatis/SqlMapConfig.xml"* />

<!--分页插件-->

<!-- <property name="plugins">

<array>

<bean class=" com.github.pagehelper.PageHelper ">

<property name="properties">

<value>

dialect=mysql

reasonable=true

</value>

</property>

</bean>

</array>

</property> -->

</bean>

<bean class=*"org.mybatis.spring.mapper.MapperScannerConfigurer"*>

<property name=*"basePackage"* value=*"com.offcn.mapper"* />

</bean>

<!--Role getRoleBySessionMember(Member member); -->

<select id=*"getRoleBySessionMember"* parameterType=*"com.offcn.pojo.Member"*

resultType=*"com.offcn.pojo.Role"*>

1. ssm整合的流程（一步一步来）

1：导包

2：整合配置文件

web.xml

*<!--加载springmvc.xml-->* <**servlet**>  
 <**servlet-name**>springmvc</**servlet-name**>  
 <**servlet-class**>org.springframework.web.servlet.DispatcherServlet</**servlet-class**>  
 <**init-param**>  
 <**param-name**>contextConfigLocation</**param-name**>  
 <**param-value**>classpath:springmvc.xml</**param-value**>  
 </**init-param**>  
 </**servlet**>  
  
<**servlet-mapping**>  
 <**servlet-name**>springmvc</**servlet-name**>  
 <**url-pattern**>/</**url-pattern**>  
</**servlet-mapping**>  
  
  
*<!--spring的监听  
springmvc: WEN-INF>>> springmvc-servlet.xml  
 spring-servlet.xml  
-->*<**context-param**>  
 <**param-name**>contextConfigLocation</**param-name**>  
 <**param-value**>classpath:spring.xml</**param-value**>  
</**context-param**>  
<**listener**>  
 <**listener-class**>org.springframework.web.context.ContextLoaderListener</**listener-class**>  
</**listener**>

springmvc.xml

*<!--springmvc的扫描：针对controller层-->*<**context:component-scan base-package="com.offcn.controller"**></**context:component-scan**>  
  
*<!--开启springmvc的配置-->* <**mvc:annotation-driven** ></**mvc:annotation-driven**>  
 <**mvc:resources mapping="/js/\*\*" location="/js/"**></**mvc:resources**>  
  
*<!--配置视图解析器-->* <**bean id="resourceViewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver"**>  
 *<!--开启前缀-->* <**property name="prefix" value="/"**></**property**>  
 *<!--开启后缀-->* <**property name="suffix" value=".jsp"**></**property**>  
 </**bean**>  
  
  
*<!--上传下载解析器-->* <**bean id="commonsMultipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver"**>  
 *<!-- 设置默认编码 -->* <**property name="defaultEncoding" value="utf-8"**></**property**>  
 *<!-- 上传图片最大大小5M-->* <**property name="maxUploadSize" value="5242440"**></**property**>  
</**bean**>

spring.xml(applicationContext.xml)

*<!--开启扫描：service的实现层-->* <**context:component-scan base-package="com.offcn.service.impl"**></**context:component-scan**>  
  
*<!--db.properties-->*<**context:property-placeholder location="classpath:db.properties"**></**context:property-placeholder**>  
*<!--datasource-->*<**bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"**>  
 <**property name="url" value="${jdbc.url}"** />  
 <**property name="username" value="${jdbc.username}"** />  
 <**property name="password" value="${jdbc.password}"** />  
 <**property name="driverClassName" value="${jdbc.driver}"** />  
</**bean**>  
  
 *<!--事务管理器-->*<**bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager"**>  
 <**property name="dataSource" ref="dataSource"**></**property**>  
</**bean**>  
  
*<!--通知tx:advice-->* <**tx:advice id="advice" transaction-manager="transactionManager"**>  
 <**tx:attributes**>  
 <**tx:method name="\*" isolation="DEFAULT" propagation="REQUIRED" read-only="false"**/>  
 </**tx:attributes**>  
</**tx:advice**>  
  
*<!--AOP-->*<**aop:config**>  
 <**aop:advisor advice-ref="advice" pointcut="execution(\* com.offcn.service.impl.PersonServiceImpl.\*(..))"**></**aop:advisor**>  
</**aop:config**>  
  
  
*<!--加载mybatis的SqlMapConfig-->* <**bean id="sqlSessionFactoryBean" class="org.mybatis.spring.SqlSessionFactoryBean"**>  
 <**property name="configLocation" value="classpath:SqlMapConfig.xml"**></**property**>  
 <**property name="dataSource" ref="dataSource"**></**property**>  
</**bean**>  
  
 <**bean class="org.mybatis.spring.mapper.MapperScannerConfigurer"**>  
 <**property name="basePackage" value="com.offcn.mapper"**></**property**>  
 </**bean**>

ssm 项目的工作流程

我们发送了一个请求，然后这个请求从我浏览器开始，一直到跳转页面，整个过程

1：发送<http://localhost:8080/ssmpro/login>?mid=1004

2:到达项目的web.xml

3:由 springmvc的DispatcherServlet映射到配置文件

springmvc.xml中

4：扫描了com.offcn.controller

5:去找@Controller》》》@RequestMapping（“login”）

6:将所携带的数据 配对 （Integer mid）

7：接受到这个mid之后交给service去处理（因为controller的任务完成了）

8：servlceimpl去将这个参数传给 mapper去进行增删改查操作

9：将增删改查操作完毕之后的数据，返回service，service返回至controller层

10：controller层根据有没有写@ReponseBody将数据打包{

a：@ReponseBody将查出来的member转化为json

b:用model，request，session封装进域中

c:实用工具类将对象转化成json

}

11：页面展示

接下来的学习方式和学习计划

上课代码必须当天完成

用最快的方式，打完你眼前的代码