SpringMVC注解和Freemarker整合使用全步骤

SpringMVC现在是比较热门的一种框架了，使用起来感觉还是很不错的，现在我分享一下集体的配置和使用，希望对学习SpringMVC的朋友有用。  
一、首先我们做准备工作，下载Spring包，下载Freemarker包。  
二、配置web.xml。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72 | <context-param>          <param-name>contextConfigLocation</param-name>          <param-value>classpath\*:/applicationContext\*.xml</param-value>      </context-param>      <filter>          <filter-name>encoding</filter-name>          <filter-class>              org.springframework.web.filter.CharacterEncodingFilter          </filter-class>          <init-param>              <param-name>encoding</param-name>              <param-value>UTF-8</param-value>          </init-param>          <init-param>              <param-name>forceEncoding</param-name>              <param-value>true</param-value>          </init-param>      </filter>      <filter-mapping>          <filter-name>encoding</filter-name>          <url-pattern>/\*</url-pattern>      </filter-mapping>      <listener>          <listener-class>              org.springframework.web.context.ContextLoaderListener          </listener-class>      </listener>      <!-- Spring 的servlet -->      <servlet>          <servlet-name>spring</servlet-name>          <servlet-class>              org.springframework.web.servlet.DispatcherServlet          </servlet-class>          <init-param>              <param-name>contextConfigLocation</param-name>              <param-value>classpath:spring-servlet.xml</param-value>          </init-param>          <load-on-startup>1</load-on-startup>      </servlet>      <servlet-mapping>          <servlet-name>spring</servlet-name>          <url-pattern>/</url-pattern>      </servlet-mapping>        <servlet-mapping>          <servlet-name>default</servlet-name>          <url-pattern>\*.jpg</url-pattern>      </servlet-mapping>      <servlet-mapping>          <servlet-name>default</servlet-name>          <url-pattern>\*.png</url-pattern>      </servlet-mapping>      <servlet-mapping>          <servlet-name>default</servlet-name>          <url-pattern>\*.js</url-pattern>      </servlet-mapping>      <servlet-mapping>          <servlet-name>default</servlet-name>          <url-pattern>\*.css</url-pattern>      </servlet-mapping>        <welcome-file-list>          <welcome-file>index.html</welcome-file>      </welcome-file-list>      <error-page>          <error-code>404</error-code>          <location>/404.jsp</location>      </error-page>      <error-page>          <error-code>500</error-code>          <location>/500.jsp</location>      </error-page> |

三、接下来配置applicationContext.xml

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 | <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans"      xmlns:aop="http://www.springframework.org/schema/aop"      xmlns:context="http://www.springframework.org/schema/context"      xmlns:p="http://www.springframework.org/schema/p"      xmlns:tx="http://www.springframework.org/schema/tx"      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"      xmlns:cache="http://www.springframework.org/schema/cache"      xsi:schemaLocation="             http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd             http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd             http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.0.xsd             http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.0.xsd             http://www.springframework.org/schema/cache http://www.springframework.org/schema/cache/spring-cache-3.1.xsd             "      default-lazy-init="true">        <description>Spring数据库及事务配置</description>        <context:component-scan base-package="com.mynote.\*,com.base.\*">          <context:exclude-filter type="annotation" expression="org.springframework.stereotype.Controller"/>      </context:component-scan>        <context:property-placeholder          location="classpath:/jdbc.properties" />        <!-- 数据库连接 -->      <bean id="dataSource"          class="org.logicalcobwebs.proxool.ProxoolDataSource">          <property name="alias">              <value>proxoolds</value>          </property>          <property name="driver">              <value>${datasource.driverClassName}</value>          </property>          <property name="driverUrl">              <value>${datasource.url}</value>          </property>          <property name="user">              <value>${datasource.username}</value>          </property>          <property name="password">              <value>${datasource.password}</value>          </property>          <property name="houseKeepingSleepTime">              <value>${datasource.houseKeepingSleepTime}</value>          </property>          <property name="maximumConnectionCount">              <value>${datasource.maximumConnectionCount}</value>          </property>          <property name="minimumConnectionCount">              <value>${datasource.minimumConnectionCount}</value>          </property>          <property name="prototypeCount">              <value>${datasource.prototypeCount}</value>          </property>          <property name="maximumActiveTime" value="1800000"/>          <property name="simultaneousBuildThrottle" value="100"/>      </bean>        <bean id="jdbcTemplate"          class="org.springframework.jdbc.core.JdbcTemplate">          <property name="dataSource" ref="dataSource" />      </bean>        <bean id="simpleJdbcInsert"          class="org.springframework.jdbc.core.simple.SimpleJdbcInsert">          <constructor-arg ref="dataSource"/>      </bean>        <!-- 事务管理 -->      <bean id="transactionManager"          class="org.springframework.jdbc.datasource.DataSourceTransactionManager">          <property name="dataSource" ref="dataSource" />      </bean>        <tx:annotation-driven transaction-manager="transactionManager" />        <bean id="ehCacheManager" class="org.springframework.cache.ehcache.EhCacheManagerFactoryBean">          <property name="shared" value="true"/>      </bean>      <bean id="ehcache" factory-bean="ehCacheManager" factory-method="getEhcache">          <constructor-arg value=""/>      </bean>        <cache:annotation-driven cache-manager="springEhCacheManager" />        <bean id="springEhCacheManager" class="com.base.modules.utils.AutoCreatedEhCacheCacheManager">          <property name="cacheManager" ref="ehCacheManager"/>      </bean>        <!-- 定期执行的业务类      <bean id="jdkExecutorJob" class="com.shst.base.bdss.bdssTask" lazy-init="false">          <property name="period" value="180" />          <property name="initialDelay" value="10" />          <property name="shutdownTimeout" value="120" />      </bean>      -->  </beans> |

四、接下来写spring-servlet.xml配置好jsp和Freemarker的自动切换，只要你的请求指定了跳转，那么就会先找对应的Freemarker文件然后会找jsp页面

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234 | <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans"      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"      xmlns:p="http://www.springframework.org/schema/p"      xmlns:context="http://www.springframework.org/schema/context"      xmlns:mvc="http://www.springframework.org/schema/mvc"      xsi:schemaLocation="http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-3.1.xsd          http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.1.xsd          http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.1.xsd">        <mvc:annotation-driven />      <!-- 将无法mapping到Controller的path交给default servlet handler处理 -->      <mvc:default-servlet-handler />        <context:component-scan base-package="com.mynote.\*"          use-default-filters="false">          <context:include-filter type="annotation"              expression="org.springframework.stereotype.Controller" />          <!--              shiro方法拦截需求              <context:include-filter type="annotation" expression="org.springframework.web.bind.annotation.ControllerAdvice" />          -->      </context:component-scan>        <mvc:interceptors>          <!-- 多个拦截器,顺序执行 -->          <mvc:interceptor>              <mvc:mapping path="/\*\*" /><!-- 如果不配置或/\*,将拦截所有的Controller -->              <bean class="com.base.modules.intercepter.SecurityIntercepter"></bean>          </mvc:interceptor>      </mvc:interceptors>        <bean          class="org.springframework.web.servlet.view.freemarker.FreeMarkerViewResolver"          p:prefix="/" p:suffix=".ftl">          <property name="cache" value="false" />          <property name="viewClass"              value="org.springframework.web.servlet.view.freemarker.FreeMarkerView" />          <property name="contentType" value="text/html;charset=UTF-8"></property>          <property name="exposeRequestAttributes" value="true" />          <property name="exposeSessionAttributes" value="true" />          <property name="exposeSpringMacroHelpers" value="true" />          <property name="requestContextAttribute" value="base"></property>          <property name="order" value="0"></property>      </bean>        <bean id="multipartResolver"          class="org.springframework.web.multipart.commons.CommonsMultipartResolver"          p:defaultEncoding="UTF-8" />        <bean id="IndexContentListTag"          class="com.mynote.tag.IndexContentListTag">      </bean>      <bean id="BKIndexContentListTag"          class="com.mynote.tag.BKIndexContentListTag">      </bean>      <bean id="contentTag" class="com.mynote.tag.ContentTag"></bean>      <bean id="HotBlogContentTag"          class="com.mynote.tag.HotBlogContentTag">      </bean>      <bean id="freemarkerConfig"          class="org.springframework.web.servlet.view.freemarker.FreeMarkerConfigurer">          <property name="templateLoaderPath" value="/templates/" />          <property name="defaultEncoding" value="UTF-8" />          <property name="freemarkerVariables">              <map>                  <entry key="indexContent"                      value-ref="IndexContentListTag">                  </entry>                  <entry key="bkContent"                      value-ref="BKIndexContentListTag">                  </entry>                  <entry key="blogContent" value-ref="contentTag"></entry>                  <entry key="hotContent" value-ref="HotBlogContentTag"></entry>              </map>          </property>          <property name="freemarkerSettings">              <props>                  <prop key="tag\_syntax">auto\_detect</prop>                  <prop key="template\_update\_delay">5</prop>                  <prop key="defaultEncoding">UTF-8</prop>                  <prop key="url\_escaping\_charset">UTF-8</prop>                  <prop key="locale">zh\_CN</prop>                  <prop key="boolean\_format">true,false</prop>                  <prop key="datetime\_format">yyyy-MM-dd HH:mm:ss</prop>                  <prop key="date\_format">yyyy-MM-dd</prop>                  <prop key="time\_format">HH:mm:ss</prop>                  <prop key="number\_format">0.######</prop>                  <prop key="whitespace\_stripping">true</prop>                  <!--空值处理<prop key="classic\_compatible">true</prop>-->                  <!--  <prop key="auto\_import">/ftl/tags/index.ftl as p,/ftl/spring.ftl as s</prop>-->              </props>          </property>      </bean>        <bean          class="org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter" />        <bean id="viewResolverCommon"          class="org.springframework.web.servlet.view.InternalResourceViewResolver">          <property name="viewClass"              value="org.springframework.web.servlet.view.JstlView" />          <property name="order" value="1"></property>          <property name="prefix" value="/WEB-INF/" />          <property name="suffix" value=".jsp"></property>      </bean>        <bean          class="org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter">          <property name="messageConverters">              <list>                    <!-- 解析json请求数据，将json转换为java对象-->                  <bean                      class="org.springframework.http.converter.json.MappingJacksonHttpMessageConverter" />                  <!-- 解析xml请求数据，将xml转换为java对象-->                    <bean                      class="org.springframework.http.converter.xml.MarshallingHttpMessageConverter">                      <constructor-arg>                          <bean                              class="org.springframework.oxm.xstream.XStreamMarshaller">                              <property name="streamDriver">                                  <bean                                      class="com.thoughtworks.xstream.io.xml.DomDriver" />                              </property>                                <property name="autodetectAnnotations">                                  <value>true</value>                              </property>                              <!--可以与xml互换的对象，需要使用XStream的注解，注解的使用方法请参XStream官网-->                              <!--                                  <property name="annotatedClasses">                                  <list>                                  <value>com.xxx.XxxxDTO</value>                                  </list>                                  </property>                              -->                          </bean>                      </constructor-arg>                  </bean>                  <bean                      class="org.springframework.http.converter.FormHttpMessageConverter" />                  <bean                      class="org.springframework.http.converter.BufferedImageHttpMessageConverter" />                  <bean                      class="org.springframework.http.converter.ByteArrayHttpMessageConverter" />                  <bean                      class="org.springframework.http.converter.StringHttpMessageConverter" />                  <!--                      <bean class="org.springframework.http.converter.ResourceHttpMessageConverter"/>                  -->                  <!--可以增加其他数据类型,请参考spring的API-->              </list>          </property>      </bean>      <bean id="fmXmlEscape"          class="freemarker.template.utility.XmlEscape" />      <bean          class="org.springframework.web.servlet.view.ContentNegotiatingViewResolver">          <property name="ignoreAcceptHeader" value="true" />          <property name="defaultContentType" value="text/html" />            <!-- 扩展名至mimeType的映射,即 /user.json => application/json -->          <property name="mediaTypes">              <map>                  <entry key="html" value="text/html" />                  <entry key="json" value="application/json" />                  <entry key="xml" value="application/xml" />              </map>          </property>          <!-- 用于开启 /userinfo/123?format=json 的支持 -->          <property name="favorParameter" value="false" />            <property name="viewResolvers">              <list>                  <bean                      class="org.springframework.web.servlet.view.BeanNameViewResolver" />              </list>          </property>          <property name="defaultViews">              <list>                  <!-- for application/json -->                  <bean                      class="org.springframework.web.servlet.view.json.MappingJacksonJsonView" />                  <!-- for application/xml -->                  <bean                      class="org.springframework.web.servlet.view.xml.MarshallingView">                      <property name="marshaller">                          <bean                              class="org.springframework.oxm.xstream.XStreamMarshaller" />                      </property>                  </bean>              </list>          </property>      </bean>        <bean id="localeResolver"          class="org.springframework.web.servlet.i18n.CookieLocaleResolver">          <property name="cookieName" value="clientlanguage" />          <property name="cookieMaxAge" value="-1" />      </bean>      <!--          <bean          class="org.springframework.aop.framework.autoproxy.DefaultAdvisorAutoProxyCreator"          depends-on="lifecycleBeanPostProcessor">          <property name="proxyTargetClass" value="true" />          </bean>            <bean          class="org.apache.shiro.spring.security.interceptor.AuthorizationAttributeSourceAdvisor">          <property name="securityManager" ref="securityManager"></property>          </bean>      -->      <bean          class="org.springframework.web.servlet.handler.SimpleMappingExceptionResolver">          <property name="exceptionMappings">              <props>                  <prop key="java.lang.Throwable">500</prop>              </props>          </property>          <property name="warnLogCategory" value="WARN"></property>          <property name="defaultErrorView" value="500"></property>          <property name="defaultStatusCode" value="500"></property>          <property name="statusCodes">              <props>                  <prop key="404">404</prop>                  <prop key="500">500</prop>              </props>          </property>        </bean>    </beans> |

五、接下来就开始写java类了，我们将使用注解来实现功能

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | @Controller  @RequestMapping  public class ContentController {        @Autowired      private TemplatesService templatesService;      @Autowired      private BlogContentService contentService;      @Autowired      private BlogInfoService blogInfoService;        @RequestMapping("{username}/{contentId}.html")      public String content(HttpServletRequest request, @PathVariable      String username, @PathVariable      String contentId) {          // 获得模版路径执行跳转          String path=templatesService.getTemplatesPath(username);          BlogInfo info = blogInfoService.findBlogInfoByUserName(username);          if(info==null){              info = new BlogInfo();              info.setName("OSblog");              info.setDes("我的博客我做主");              info.setNickname("Osblog博客");          }          request.setAttribute("bloginfo", info);          contentService.updateclicks(contentId);          request.setAttribute("bloguser", username);          BlogContent content=contentService.getEntityById(contentId);          request.setAttribute("content", content);          return path+"/content";      }  } |

这是一个控制器，和Struts中Action是一样的，上面的方法是动态接受请求的方法，比如请求是http://localhost:8080/osblog/12.html,在控制器中取到的username=osblog,contentid=12,是不是很智能啊。  
剩下的数据层和业务层写法就没有什么可写的了，注解的方式也和普通注解一样。在配置文件中写到了：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | <bean          class="org.springframework.web.servlet.view.freemarker.FreeMarkerViewResolver"          p:prefix="/" p:suffix=".ftl">          <property name="cache" value="false" />          <property name="viewClass"              value="org.springframework.web.servlet.view.freemarker.FreeMarkerView" />          <property name="contentType" value="text/html;charset=UTF-8"></property>          <property name="exposeRequestAttributes" value="true" />          <property name="exposeSessionAttributes" value="true" />          <property name="exposeSpringMacroHelpers" value="true" />          <property name="requestContextAttribute" value="base"></property>          <property name="order" value="0"></property>      </bean> |

这个就是指明了你调转的视图是Freemarker的，同理另一个指明的是jsp按照顺序他们会自动选这你要跳转的视图。不过在注解是@RequestMapping("sitemap.html")里面的值绝对不能重复的。