**一、编写spring-datasource.xml集合mybatis和sping。**

<?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"

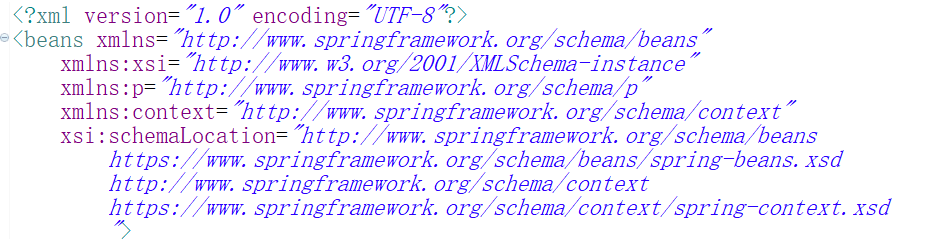
xsi:schemaLocation="http://www.springframework.org/schema/beans

https://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

https://www.springframework.org/schema/context/spring-context.xsd

">



**1、配置数据源：**

<!-- 1，加载数据信息属性文件 -->

<context:property-placeholder location="classpath:jdbc.properties"/>

<!-- 2,数据源：数据库的相关信息 :官网有 bdcp数据源，是数据源的一种 -->

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">

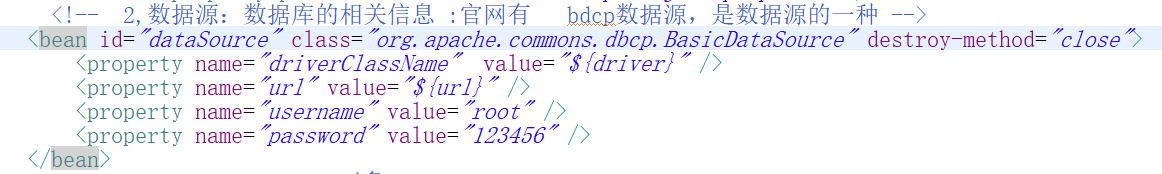
<property name="driverClassName" value="${driver}" />

<property name="url" value="${url}" />

<property name="username" value="root" />

<property name="password" value="123456" />

</bean>



**2、配置SqlSessionFactory对象**

<!-- 3、sqlSessionFactory 对象 -->

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

<!-- 别名 -->

<property name="typeAliasesPackage" value="com.dy.model"></property>

<!-- mapper XML映射 -->

<property name="mapperLocations" value="classpath\*:mappers/\*Mapper.xml"></property>

<!-- 数据源 -->

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

<!-- 分页插件 -->

<property name="plugins">

<array>

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

<property name="properties">

<!--使用下面的方式配置参数，一行配置一个 -->

<value>

helperDialect=mysql

</value>

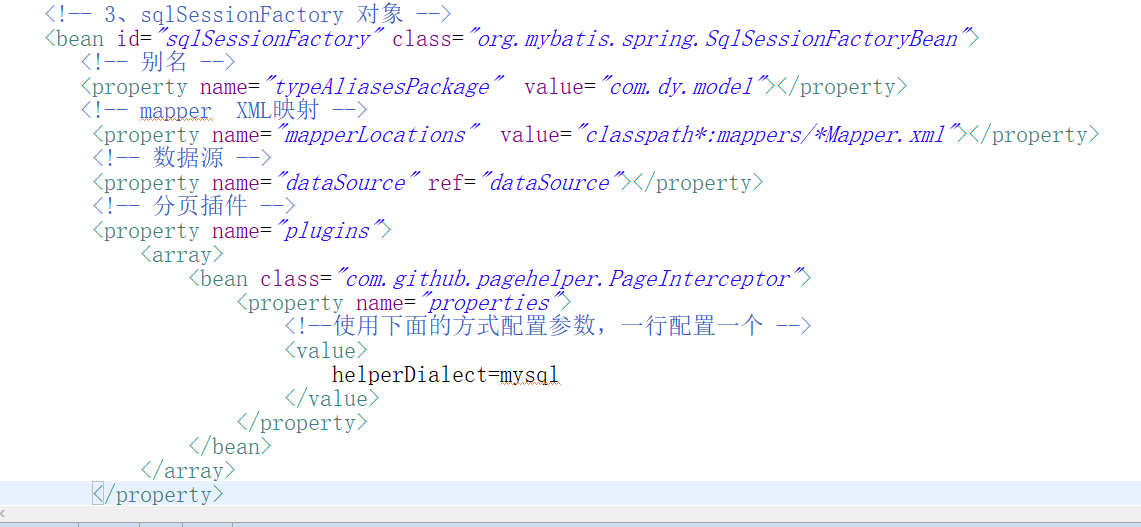
</property>

</bean>

</array>

</property>

</bean>



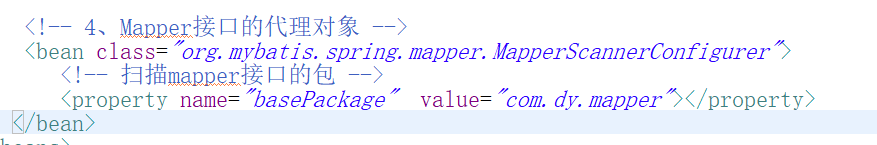
**3、配置Mapper接口代理对象**

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

<!-- 扫描mapper接口的包 -->

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

</bean>



**二、配置baeans扫描spring-datasource.xm。**

<?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"

xsi:schemaLocation="http://www.springframework.org/schema/beans

https://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

https://www.springframework.org/schema/context/spring-context.xsd

">



配置注解扫描，扫描文件

<!-- 1、注解扫描 -->

<context:component-scan base-package="com.dy.\*\*">

<!-- 排除@Controller注解 -->

<context:exclude-filter type="annotation" expression="org.springframework.stereotype.Controller"/>

</context:component-scan>

<!-- 2、datasource:mybatis -->

<import resource="spring-datasource.xml"/>

<!-- 3、事务管理器 -->

**三、配置中央处理器spring-web.xml文件**

<?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/beans

https://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

https://www.springframework.org/schema/context/spring-context.xsd

http://www.springframework.org/schema/mvc

https://www.springframework.org/schema/mvc/spring-mvc.xsd

">

**1、扫描注解**

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

clipboard.png

**2、配置适配器，将json配置放入里面**

<bean

class="org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter">

<property name="messageConverters">

<list>

<bean class="org.springframework.http.converter.StringHttpMessageConverter"/>

<bean

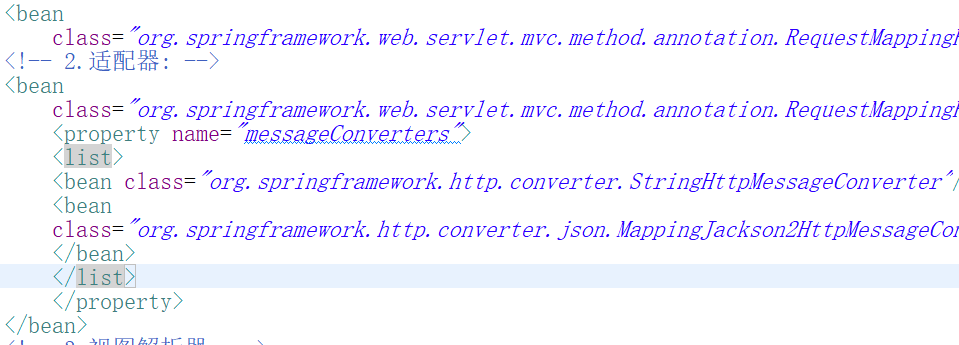
class="org.springframework.http.converter.json.MappingJackson2HttpMessageConverter">

</bean>

</list>

</property>

</bean>



**3、配置视图解析器**

<bean

class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<!-- 解析jstl标签 -->

<property name="viewClass"

value="org.springframework.web.servlet.view.JstlView" />

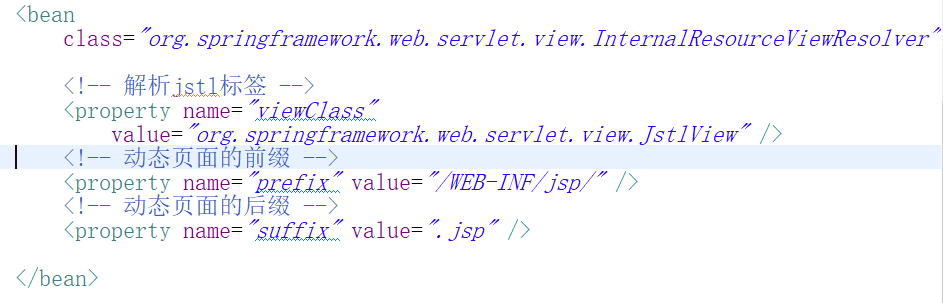
<!-- 动态页面的前缀 -->

<property name="prefix" value="/WEB-INF/jsp/" />

<!-- 动态页面的后缀 -->

<property name="suffix" value=".jsp" />

</bean>



四、web.xml文件扫描spring-web.xml



<display-name>nd-springmvc01-base</display-name>

<servlet>

<servlet-name>springmvc</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<!-- 参数 :告诉中央处理器 使用哪个映射器，哪个适配器，哪个视图解析器-->

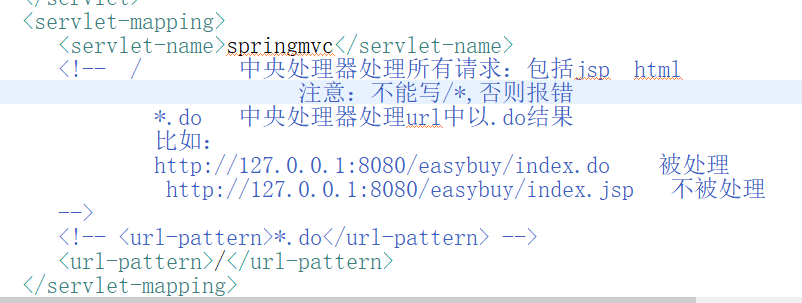
<init-param>

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

<param-value>classpath:spring-web.xml</param-value>

</init-param>

</servlet>



<servlet-mapping>

<servlet-name>springmvc</servlet-name>

<!-- / 中央处理器处理所有请求：包括jsp html

注意：不能写/\*,否则报错

\*.do 中央处理器处理url中以.do结果

比如：

http://127.0.0.1:8080/easybuy/index.do 被处理

http://127.0.0.1:8080/easybuy/index.jsp 不被处理

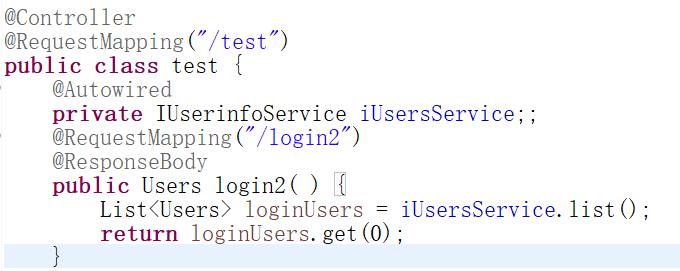
-->

<!-- <url-pattern>\*.do</url-pattern> -->

<url-pattern>/</url-pattern>

</servlet-mapping>

五、通过@RequestBody和@ResponseBody编写接口



六、使用Postman测试

