SpringMVC学习笔记

1. HelloWord实例
2. 在lib目录下添加jar包
3. 在src源目录下添加spring-mvc.xml配置文件

<!-- 自动扫面包下使用了注解的类，包括子集，装载bean -->

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

<!-- 视图解析器 -->

<bean id="viewResolver"

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

<!-- 会在value指定的路径下找指定方法返回值拼接.jsp后的文件 -->

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

/>

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

</bean>

1. 在web.xml文件下配置springmvc的映射关系以及一些基本配置

<!-- 配置SpringMVC -->

<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:spring-mvc.xml</param-value>

</init-param>

</servlet>

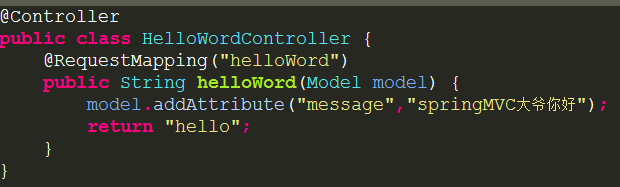
<servlet-mapping>

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

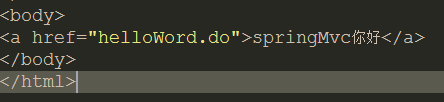
<url-pattern>\*.do</url-pattern>

</servlet-mapping>

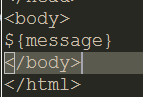
1. 编写controller层



1. 编写访问该controller的页面文件

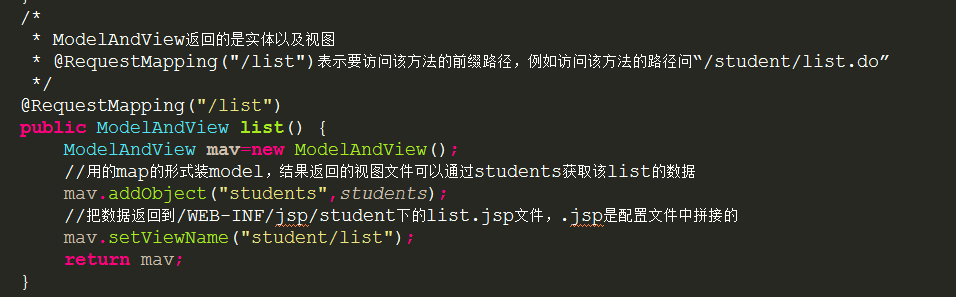


1. 编写controller返回结果的视图文件hello.jsp

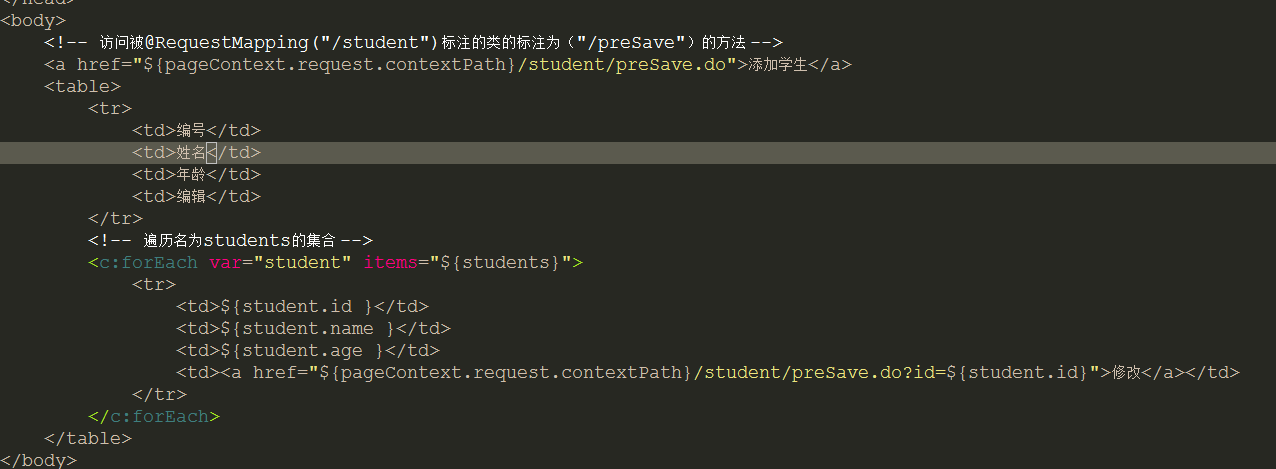


1. SpringMVC控制器w

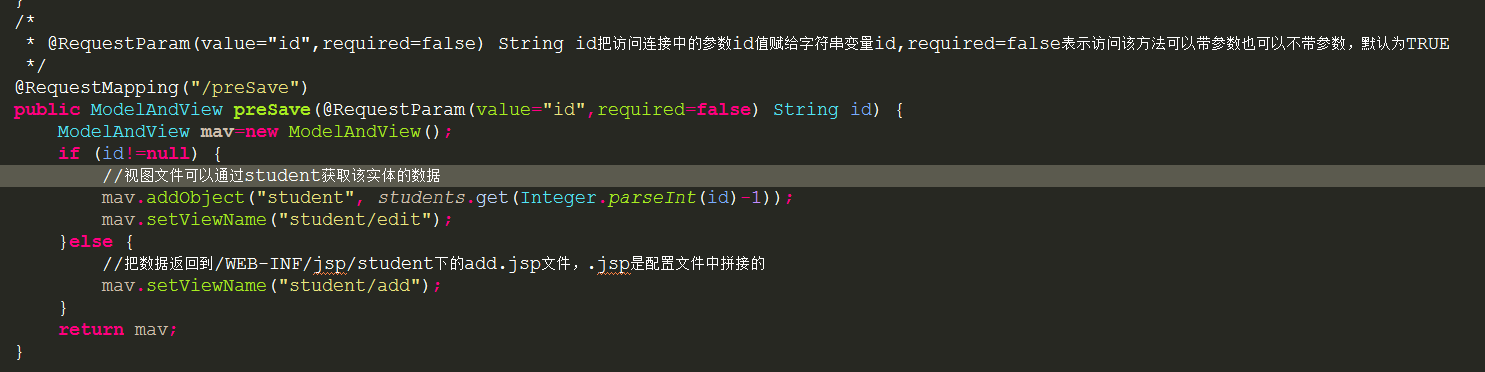
第一节：@RequestMapping请求映射

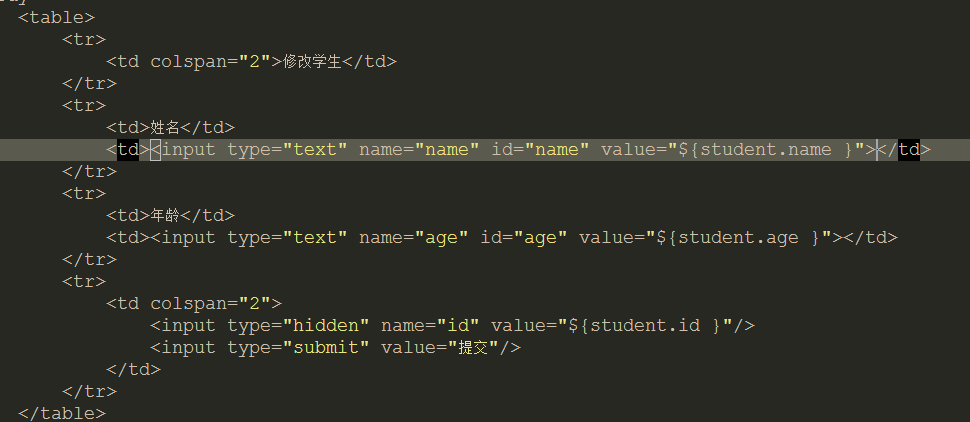


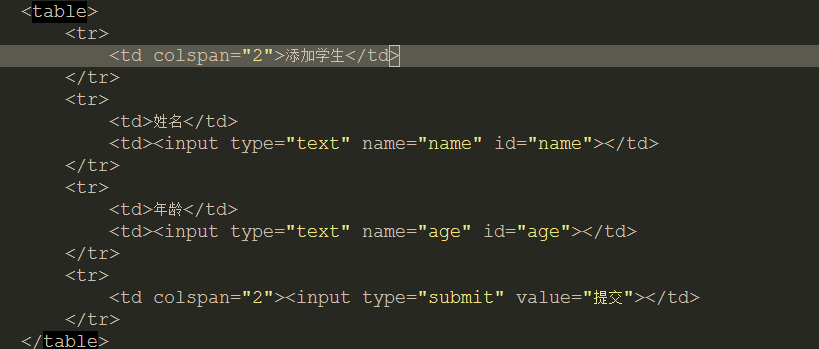
第二节：@RequestParma请求参数



第三节：返回模型和视图







第四节：SpringMVC对象自动封装属性

第五节：请求乱码的解决，在web.xml中加入过滤器

<!-- 该过滤器是解决中文乱码问题，转为utf-8编码格式 -->

<filter>

<filter-name>characterEncodingFilter</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>characterEncodingFilter</filter-name>

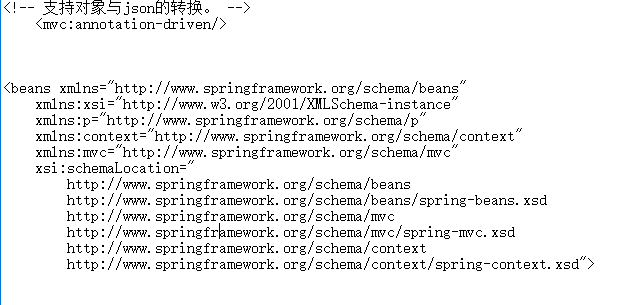
<url-pattern>\*.do</url-pattern>

</filter-mapping>

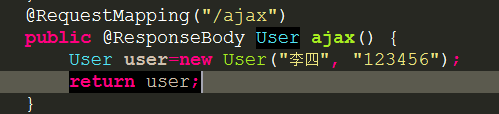
第七节：对ServletAPI的支持

第八节：对json的支持

添加json所需jar包以及声明配置，然后使用@ResponseBody注解，该注解用于将Controller的方法返回的对象，通过适当的HttpMessageConverter转换为指定格式后，写入到Response对象的body数据区。

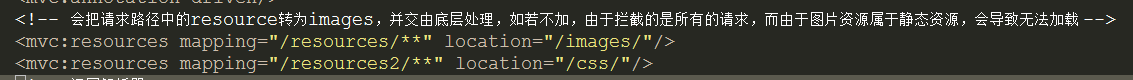


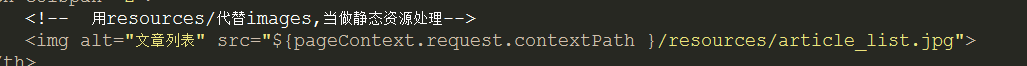
返回的user是一个json格式



第四章：Restful风格的资源URL









第四章：SpringMVC文件的上传

单文件上传以及多文件上传，首先要配置一个文件上传需要用到的bean

<!-- springMvc文件上传需要用到的bean -->

<bean id="multipartResolver"

class="org.springframework.web.multipart.commons.CommonsMultipartResolver">

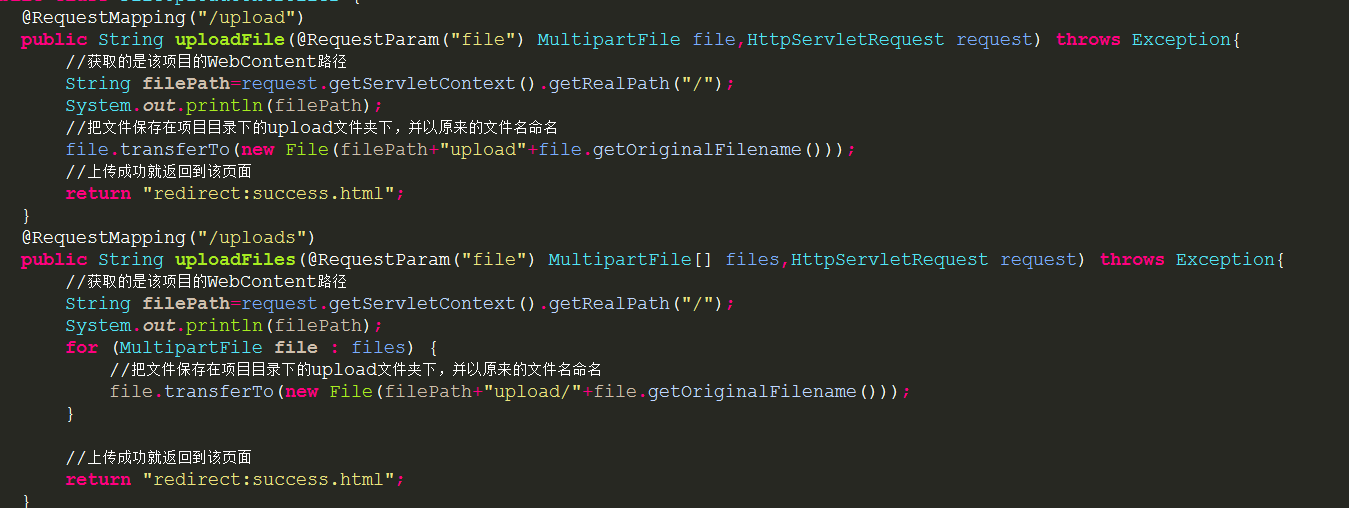
<!-- 设置编码格式 -->

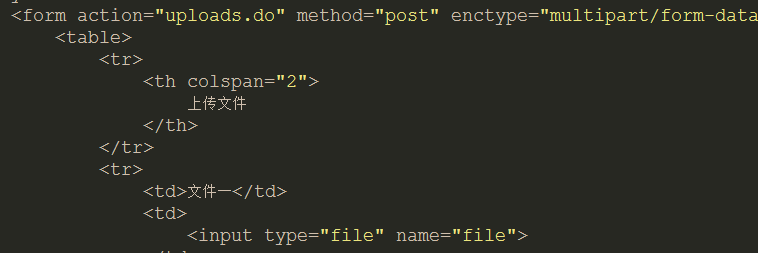
<property name="defaultEncoding" value="UTF-8"/>

<!-- 设置最大文件上传字节 -->

<property name="maxUploadSize" value="10000000"/>

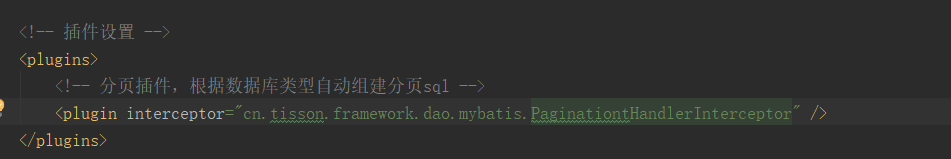
</bean>

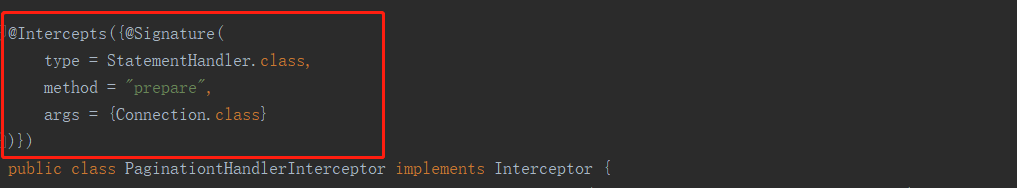




Plugins插件的使用

在mybatis.xml的配置文件中使用<plugins>标签，自定义一个拦截器并继承Interceptor





在自定义的拦截器上需要定义拦截的类，拦截的方法，以及参数，具体百度

这样就可以在已映射语句执行过程中的某一点进行拦截调用。

例如：

把Mybatis所有执行的sql都记录下来

拦截sql语句，自动加上分页的语法