**1、常用xml配置**

**<!-- 扫描包里的注解自动创建bean -->**

<context:component-scan base-package="com.cdsxt.test.testMvcController"/>

**<!-- 开启支持扫描mvc注解 -->**

<mvc:annotation-driven />

**<!-- jsp视图解析器，带前后缀配置 -->**

<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver"

id="internalResourceViewResolver">

<property name="prefix" value="/testJsp/" />

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

</bean>

**<!-- spring mvc 的拦截器 -->**

<mvc:interceptors > </ mvc:interceptors >

**<!-- SpringMVC上传文件时，需要配置MultipartResolver处理器 -->**

**<!-- CommonsMultipartResolver MultipartResolver的一个实现，需要commons-fileupload 等 jar 包支持， 但它能在比较旧的 servlet 版本中使用 -->**

<bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver">

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

<property name="maxUploadSize" value="10485760000" />

<property name="maxInMemorySize" value="40960" />

</bean>

**2、常用anno注解配置**

@Controller

　　注解到控制器类上，创建对应的对象到spring 上下文中

@RequestMapping

　　注解为控制器类或方法上，进行url映射

a）直接注解到【控制类的处理方法】上

b）注册到【控制类】和【控制类的处理方法】上

------------------------处理方法-请求参数--------------------------------------

@RequestParam　（http请求参数）

如：http://域名:端口/文件路径?name=zengxiaoxian&age=18

请求头

请求body

　　在【处理方法】形式参数前，使用 @RequestParam 可以把http请求的参数注入给

对应的形式参数（类型：基本类型，String，Map<String, String>，javaBean对象）

@RequestBody（http请求body）

如：http://域名:端口/文件路径

请求头

请求body：这里的数据会被RequestBody取出来

　　在【处理方法】形式参数前，使用 @RequestParam 可以把http请求的body注入给对应的形式参数，使用系统默认配置的HttpMessageConverter进行解析转换，

@PathVariable（http请求路径）

　　绑定 URL 占位符到入参

-----------------------处理方法-返回值--------------------------------------

@ResponseBody

　　 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

a）使用了直接把返回的内容-转换为字符串-》写出body数据区

b）没有使用则通过返回的名字找到对应的视图解析器进行解析，解析后把html返回

@ModelAttribute

　　在方法定义上使用 @ModelAttribute 注解：Spring MVC 在调用目标处理方法前，会先逐个调用在方法级上标注了@ModelAttribute 的方法

　　在方法的入参前使用 @ModelAttribute 注解：可以从隐含对象中获取隐含的模型数据中获取对象，再将请求参数 –绑定到对象中，再传入入参将方法入参对象添加到模型中

@ExceptionHandler

　　注解到方法上，出现异常时会执行该方法

@ControllerAdvice

　　使一个Contoller成为全局的异常处理类，类中用@ExceptionHandler方法注解的方法可以处理所有Controller发生的异常