1. SpringMVC自带一个DispatchherServlet，可以把它配置在部署描述文件中。

<servlet>  
 <servlet-name>dispatcherServlet</servlet-name>  
 <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  
 <load-on-startup>1</load-on-startup>  
</servlet>  
<servlet-mapping>  
 <servlet-name>dispatcherServlet</servlet-name>  
 <url-pattern>/</url-pattern>  
</servlet-mapping>

1. Dispatcher servlet初始化时，将寻找一个在应用程序的WEB-INF目录下的配置文件：servletName-servlet.xml，如上诉配置的servlet对应的文件名是dispatcherServlet-servlet.xml

也可以把这个文件放在任何地方，可以使用init-param元素指定此文件路径：

<init-param>  
 <param-name>contextConfigLocation</param-name>  
 <param-value>/WEB-INF/config/dispatcherServlet.xml</param-value>  
</init-param>

1. 视图解析器View Resolver
   1. InternalResourceViewResolver 解析资源路径（不用再写前缀和后缀）

<bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">  
 <property name="prefix" value="/WEB-INF/jsp/" />  
 <property name="suffix" value=".jsp" />  
</bean>

1. 使用@Controller注解声明控制器
   1. 在spring mvc配置文件中声明spring-context
   2. 添加component-scan元素

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

* 1. 使用@RequestMapping设置请求路径和方法。
  2. 需要什么就把什么作为参数，比如HttpSession，HttpServletRequest等

可以出现的参数类型如下：

•ServletRequest、HttpServletRequest

•ServletResponse、HttpServletResponse

•HttpSession

•WebRequest、NativeWebRequest

•Locale

•InputStream、Reader

•OutputStream、Writer

•Principal

••HttpEntity<?>

•Map、Model：Model是一个包含Map的Spring MVC类型，在每次调用请求处理方法时，Spring MVC都创建Model对象并将各种对象注入到Map中

•RedirectAttributes

•Errors

•BindingResult

命令或表单对象：

•SessionStatus

•UriComponentBuilder

•带@PathVariable，@MatrixVariable注释的对象

•@RequestParam,@RequestHeader，@RequestBody，@RequestPart

可以返回的对象：

•ModelAndView

•Model

•Map包含模型的属性

•View

•代表逻辑视图名的String

•Void

•提供对Servlet的访问，以响应HTTP头部和内容HttpEntity或ResponseEntity对象

•Callable

•DeferredResult

•其他任意类，Spring将其视作输出给View的对象模型。

1. 又有控制器又有静态页面的配置方法，在spring mvc里面配置下面内容：

<mvc:annotation-driven/><!-- 注册基于注解的控制器处理方法的bean对象，如果要用resources标签则一定要由，否则控制器将被阻止调用 -->  
<mvc:resources mapping="/css/\*\*" location="/css/" /> <!-- 配置/css目录下的所有文件可见 -->  
<mvc:resources mapping="/\*.html" location="/" /><!-- 配置.html文件可见 -->

1. 重定向和Flash属性
   1. 转发比重定向快，因为重定向经过客户端。若需要重定向到一个外部网站，则无法使用转发。
   2. 要使用Flash属性必须在Spring MVC配置文件中添加<annotation-driven/>元素，还必须在方法上添加RedirectAttributes参数
2. @ModelAttribute带@ModelAttribute注解的方法或者参数会将参数对象添加到Model对象中

@RequestMapping("test")  
@ResponseBody  
public String test(@ModelAttribute("name") String name, Model model) {  
 System.*out*.println(model.asMap().get("name"));  
 return "teetetes";  
}

如果方法带@ModelAttribute注解，会在该控制器类的请求处理方法被调用时调用此方法，返回值会自动添加到Model里（如果返回值为void则必须添加一个Model类型的参数，并自行将示例添加到Model中）

@ModelAttribute  
public String test(String name) {  
 return "teetetes";  
}