**一：SpringMVC是什么：**

　　1.**[spring](http://lib.csdn.net/base/javaee" \o "Java EE知识库" \t "http://blog.csdn.net/changhenshui1990/article/details/_blank)** MVC是一个MVC的开源框架， **[spring](http://lib.csdn.net/base/javaee" \o "Java EE知识库" \t "http://blog.csdn.net/changhenshui1990/article/details/_blank)** MVC属于SpringFrameWork的后续产品，spring MVC=struts2+spring，springMVC就相当于是Struts2加上sring的整合，但是这里有一个疑惑就是，springMVC和spring是什么样的关系呢？这个在百度百科上有一个很好的解释：意思是说，springMVC是spring的一个后续产品，其实就是spring在原有基础上，又提供了web应用的MVC模块，可以简单的把springMVC理解为是spring的一个模块（类似AOP，IOC这样的模块），网络上经常会说springMVC和spring无缝集成，其实springMVC就是spring的一个子模块，所以根本不需要同spring进行整合。

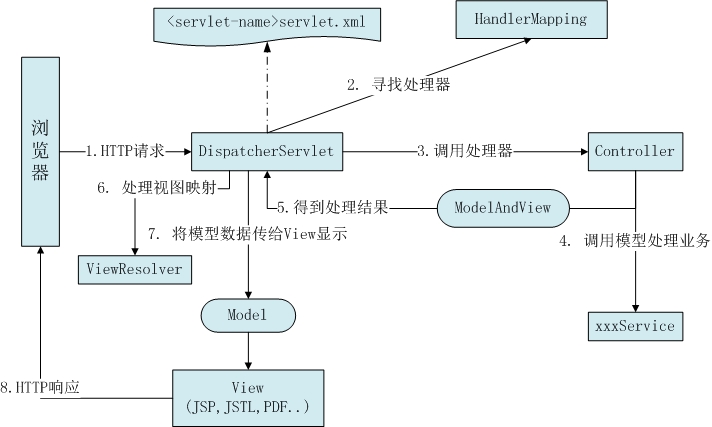
      2.Spring MVC 分离了控制器、模型对象、分派器以及处理程序对象的角色，这种分离让它们更容易进行定制。

      3.Spring的MVC框架主要由DispatcherServlet、处理器映射、处理器(控制器)、视图解析器、视图组成。

      4.他的两个核心是两个核心：

处理器映射：选择使用哪个控制器来处理请求   
视图解析器：选择结果应该如何渲染

# 二：SpringMVC原理图

****

# 三：SpringMVC运行原理

# ****Http请求****：客户端请求提交到DispatcherServlet。  ****寻找处理器****：由DispatcherServlet控制器查询一个或多个HandlerMapping，找到处理请求的Controller。  ****调用处理器****：DispatcherServlet将请求提交到Controller。  ****调用业务处理和返回结果****：Controller调用业务逻辑处理后，返回ModelAndView。  ****处理视图映射并返回模型****： DispatcherServlet查询一个或多个ViewResoler视图解析器，找到ModelAndView指定的视图。

**Http响应**：视图负责将结果显示到客户端。

**四：Spring mvc 组件**

1.前端控制器（DispatcherServlet）（不需要工程师开发）,由框架提供

作用：接收请求，响应结果，相当于转发器，中央处理器。有了dispatcherServlet减少了其它组件之间的耦合度。  
用户请求到达前端控制器，它就相当于mvc模式中的c，dispatcherServlet是整个流程控制的中心，由它调用其它组件处理用户的请求，dispatcherServlet的存在降低了组件之间的耦合性。

2.处理器映射器（HandlerMapping）（不需要工程师开发）,由框架提供

作用：根据请求的url查找Handler  
HandlerMapping负责根据用户请求找到Handler即处理器，springmvc提供了不同的映射器实现不同的映射方式，例如：配置文件方式，实现接口方式，注解方式等。

3.处理器（Handler）：（需要程序员去写代码处理逻辑的）

作用：Handler 是继DispatcherServlet前端控制器的后端控制器，在DispatcherServlet的控制下Handler对具体的用户请求进行处理。  
由于Handler涉及到具体的用户业务请求，所以一般情况需要工程师根据业务需求开发Handler。

4.处理器适配器（HandlerAdapter）

作用：会把处理器包装成适配器，这样就可以支持多种类型的处理器，类比笔记本的适配器（适配器模式的应用）

5.视图解析器（ViewResovler）(不需要工程师开发),由框架提供

作用：进行视图解析，根据逻辑视图名解析成真正的视图（view）  
View Resolver负责将处理结果生成View视图，View Resolver首先根据逻辑视图名解析成物理视图名即具体的页面地址，再生成View视图对象，最后对View进行渲染将处理结果通过页面展示给用户。 springmvc框架提供了很多的View视图类型，包括：jstlView、freemarkerView、pdfView等。  
一般情况下需要通过页面标签或页面模版技术将模型数据通过页面展示给用户，需要由工程师根据业务需求开发具体的页面。

6、视图View(需要工程师开发jsp...)  
View是一个接口，实现类支持不同的View类型（jsp、freemarker、pdf...）

**五：SpringMVC接口解释：**

1.DispatcherServlet接口： Spring提供的前端控制器，所有的请求都有经过它来统一分发。在DispatcherServlet将请求分发给Spring

2.HandlerMapping接口： 能够完成客户请求到Controller映射。

3.Controller接口： 需要为并发用户处理上述请求，因此实现Controller接口时，必须保证线程安全并且可重用。 Controller将处理用户请求，这和Struts Action扮演的角色是一致的。一旦Controller处理完用户请求，则返回ModelAndView对象给DispatcherServlet前端控制器，ModelAndView中包含了模型（Model）和视图（View）。

从宏观角度考虑，DispatcherServlet是整个Web应用的控制器；从微观考虑，Controller是单个Http请求处理过程中的控制器，而ModelAndView是Http请求过程中返回的模型（Model）和视图（View）。

4.ViewResolver接口： Spring提供的视图解析器（ViewResolver）在Web应用中查找View对象，从而将相应结果渲染给客户。

**六：DispatcherServlet是整个Spring MVC的核心。**

它负责接收HTTP请求组织协调Spring MVC的各个组成部分。

其主要工作有以下三项：

1、截获符合特定格式的URL请求。

2、初始化DispatcherServlet上下文对应的WebApplicationContext，并将其与业务层、持久化层的WebApplicationContext建立关联。

3、初始化Spring MVC的各个组成组件，并装配到DispatcherServlet中。

注明：Spring MVC Controller默认是单例的：

单例的原因有二：

1、为了性能。

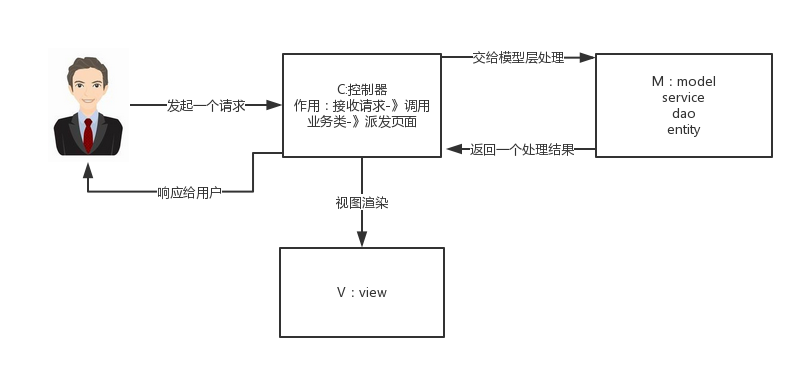
2、不需要多例（方法级别，没有共享属性的情况下，线程安全的。）

3、SpringIOC默认注入就是单例的。

**七：MVC模型结构**

MVC：MVC是一种设计模式

MVC的原理图：



分析：

M-Model 模型（完成业务逻辑：有javaBean构成，service+dao+entity）

V-View 视图（做界面的展示  jsp，html……）

C-Controller 控制器（接收请求—>调用模型—>根据结果派发页面）