参看文档：https://blog.csdn.net/eson\_15/article/details/51689023

# Spring MVC框架

Spring 为展现层提供的基于 MVC 设计理念的优秀的Web 框架，是目前最主流的 MVC 框架之一

Spring3.0 后全面超越 Struts2，成为最优秀的 MVC 框架

Spring MVC 通过一套 MVC 注解，让 POJO 成为处理请求的控制器，而无须实现任何接口。

支持 REST 风格的 URL • 请求

采用了松散耦合可插拔组件结构，比其他 MVC 框架更具扩展性和灵活性

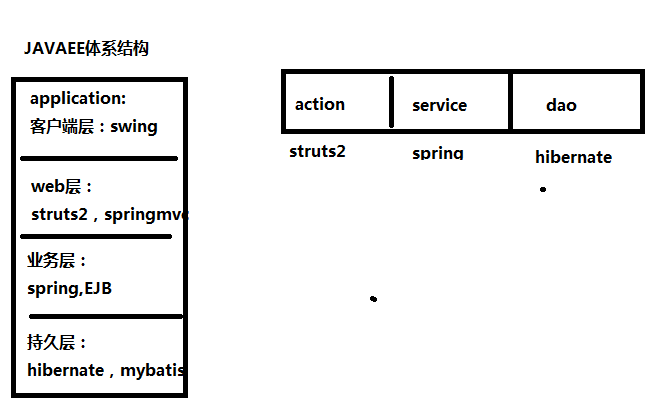
## Spring mvc介绍

Spring web mvc和Struts2都属于表现层的框架,它是Spring框架的一部分,我们可以从Spring的整体结构中看得出来：



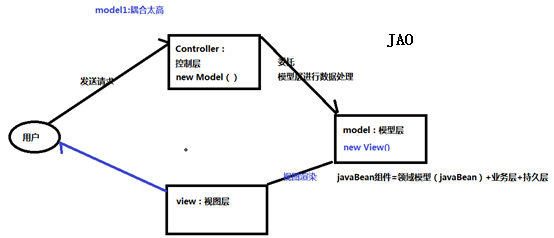
## mvc与jee架构

### JAVAEE体系结构

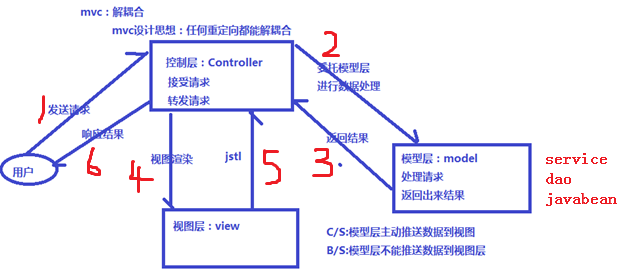


### MVC

#### Model1：



#### Model2：



mvc设计模式在b/s系统下应用：

1发起请求

C 控制器controller

作用：接收请求，响应处理结果，没有业务逻辑

5 响应结果

M 模型

pojo

service

dao

2请求业务处理

V 视图 View

将模型数据展示给用户

4 向view中填充模型数据

3返回模型

注意：b/s下模型无法将数据设置到视图中

1. 用户发起request请求至控制器(Controller)

控制接收用户请求的数据，委托给模型进行处理

1. 控制器通过模型(Model)处理数据并得到处理结果

模型通常是指业务逻辑

1. 模型处理结果返回给控制器
2. 控制器将模型数据在视图(View)中展示

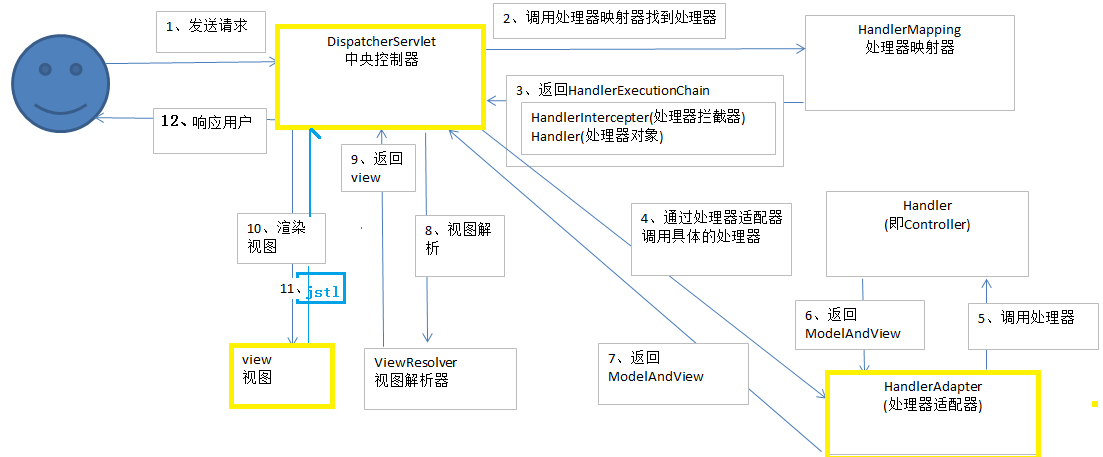
web中模型无法将数据直接在视图上显示，需要通过控制器完成。如果在C/S应用中模型是可以将数据在视图中展示的。

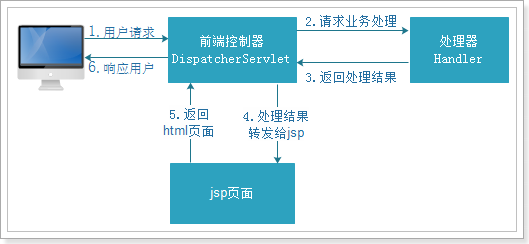
1. 控制器将视图response响应给用户

通过视图展示给用户要的数据或处理结果。

## Spring mvc 架构

### 架构图





### 架构流程

第一步：用户发起request请求，请求至DispatcherServlet前端控制器(中央控制器)

第二步：DispatcherServlet前端控制器请求HandlerMapping处理器映射器查找Handler

第三步：HandlerMapping处理器映射器，根据url及一些配置规则（xml配置、注解配置）查找Handler，将Handler返回给DispatcherServlet前端控制器

第四步：DispatcherServlet前端控制器调用适配器执行Handler，有了适配器通过适配器去扩展对不同Handler执行方式（比如：原始servlet开发，注解开发）

第五步：适配器执行Handler

Handler是后端控制器，当成模型。

第六步：Handler执行完成返回ModelAndView

ModelAndView：springmvc的一个对象，对Model和view进行封装。

第七步：适配器将ModelAndView返回给DispatcherServlet

第八步：DispatcherServlet调用视图解析器进行视图解析，解析后生成view

视图解析器根据逻辑视图名解析出真正的视图。

View：springmvc视图封装对象，提供了很多view，jsp、freemarker、pdf、excel。。。

第九步：ViewResolver视图解析器给前端控制器返回view

第十步：DispatcherServlet调用view的渲染视图的方法，将模型数据填充到request域 。

第十一步：jstl通过中央处理器

第十二步：DispatcherServlet向用户响应结果(jsp页面、json数据。。。。)

### 组件说明

以下组件通常使用框架提供实现，大都springmvc提供，写在springmvc.xml中：

DispatcherServlet：前端控制器

又名 中央控制器

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。

HandlerMapping：处理器映射器

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。

Handler：处理器

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。

HandlAdapter：处理器适配器

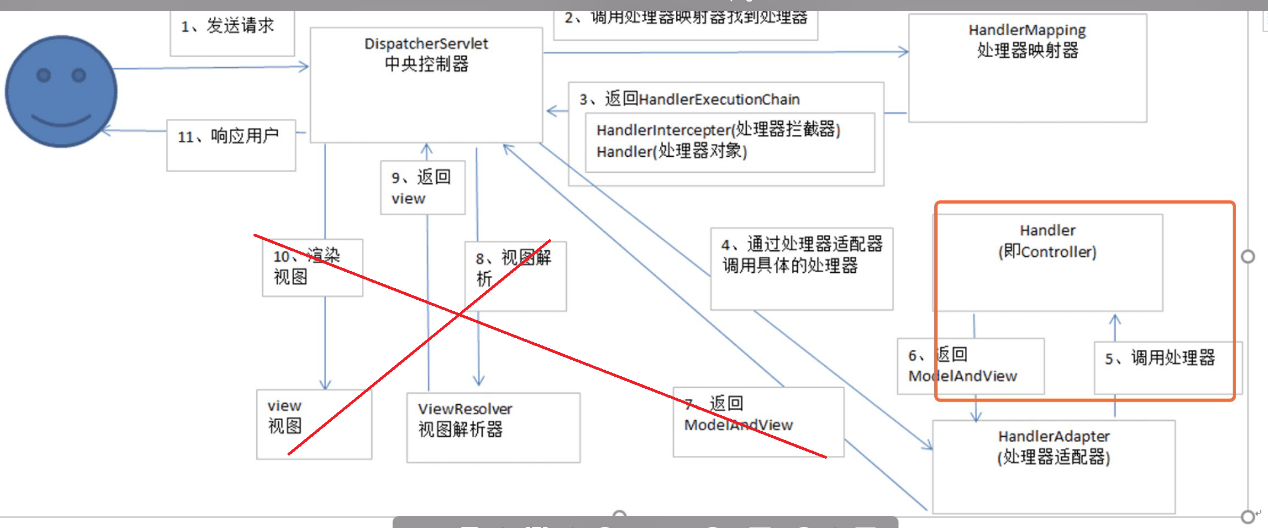
通过HandlerAdapter对处理器进行执行，这是适配器模式的应用，通过扩展适配器可以对更多类型的处理器进行执行。

ViewResolver：视图解析器

ViewResolver负责将处理结果生成View视图，ViewResolver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。

## Spring MVC架构-json

### 架构图



### Json数据处理

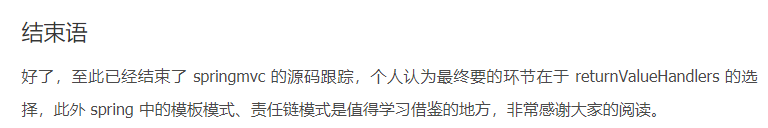
https://blog.csdn.net/thewindkee/article/details/80517703

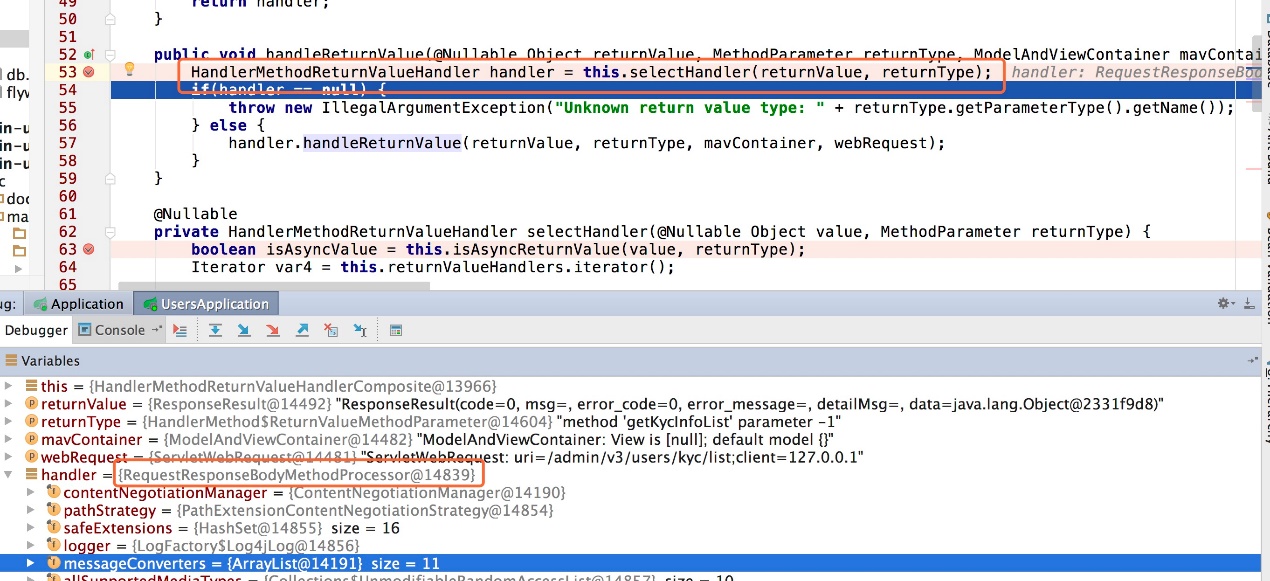


json是直接就返回数据了，不用走后面找视图模版（jsp,freemarker），用model数据渲染页面的逻辑

### 原理

https://www.infoq.cn/article/ahOwqr2Dtfg6mgQRxhET





# struts2和springmvc？

1. 运行速度：

Struts2是基于filter过滤器实现的。

Springmvc基于servlet实现（即前端控制器）。Servlet比过滤器快。

2. 入口实现机制：

struts2是基于类开发，传递参数是通过类的属性，只能设计为多例。

eg: 请求来了以后，struts2创建多少个对象：ActionContext，valuestack，UserAction，ActionSuport，ModelDriven

userAction里面属性：User对象，userlist集合等

springmvc是基于方法开发(一个url对应一个方法)，请求参数传递到方法的形参，可以设计为单例或多例(建议单例)

3.参数封装来分析：

Struts基于属性封装（采用值栈存储请求和响应的数据，通过OGNL存取数据）。

Springmvc基于方法封装（通过参数解析器是将request请求内容解析，并给方法形参赋值，将数据和视图封装成ModelAndView对象，最后又将ModelAndView中的模型数据通过request域传输到页面）。颗粒更细Jsp视图解析器默认使用jstl。

