### 依赖注入有哪几种注解

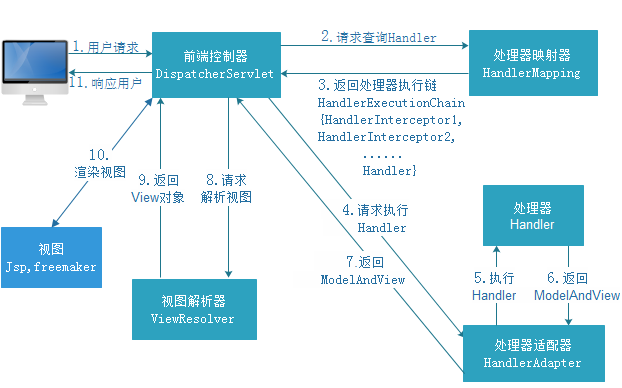
@Autowired

@Resource

### Spring实现事务的方式有哪些

### SpringMVC的执行流程

#### 1.解释一



一个请求匹配前端控制器 DispatcherServlet 的请求映射路径(在 web.xml中指定), WEB 容器将该请求转交给 DispatcherServlet 处理

DispatcherServlet 接收到请求后, 将根据 请求信息 交给 处理器映射器 （HandlerMapping）

HandlerMapping 根据用户的url请求 查找匹配该url的 Handler，并返回一个执行链

DispatcherServlet 再请求 处理器适配器(HandlerAdapter) 调用相应的 Handler 进行处理并返回 ModelAndView 给 DispatcherServlet

DispatcherServlet 将 ModelAndView 请求 ViewReslover（视图解析器）解析，返回具体 View

DispatcherServlet 对 View 进行渲染视图（即将模型数据填充至视图中）

DispatcherServlet 将页面响应给用户

##### 组件说明：

###### DispatchServlet：前端控制器

用户请求到达前端控制器，相当于MVC模式中的C，dispatchServlet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，dispatchServlet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url找到HandlerMapping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DispatchServlet前端控制器的后端控制器，在DispatchServlet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，所以一般情况需要程序员根据业务需求开发Handler。

###### HandlerAdapter：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

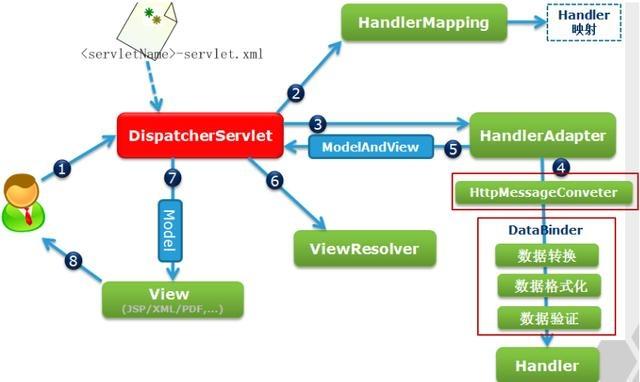
ViewResolver负责将处理结果生成view视图，ViewResolver首先根据逻辑视图名解析成物理视图名即具体的页面地址。再生成view视图对象，最后对view进行渲染将处理结果通过页面展示给用户。

###### View：视图

SpringMVC框架提供很多的view视图类型的支持，包括：jstlView、freemarkerView、pdfView等，一般常用的视图就是jsp。

一般情况下需要通过页面标签或页面模板技术奖模型数据通过页面展示给用户，需要由程序员根据业务需求开发具体的页面。

#### 2.解释二



### SpringMVC常用注解

@Controller：此类为控制器

@RequestMapping：给控制器方法绑定一个url

@ResponseBody：将java对象转换成json，并且发送给客户端

@RequestBody：将客户端请求过来的json转换成java对象

@RequestParam：当表单参数和方法形参名字不一致时，做一个名字映射

@PathVarible：用于获取url中的参数，比如user/1中1的值

#### Rest风格的新api

@RestController：@Controller+@ResponseBody

@GetMapping @DeleteMapping @PostMapping @PutMapping

#### 其他注解

@SessionAttribute：声明将什么模型数据存入session

@CookieValue：获取cookie值

@ModelAttribute：将方法返回值存入model中

@HeaderValue：获取请求头的值

### SpringMVC与Struts2的区别

#### 1.入口不同

SpringMVC入口是Servlet；Struts2入口是filter

#### 2.生命周期不同

SpringMVC controller是单例的，不能使用成员变量获取参数，所以效率高

Struts2 action是多例的，可以使用成员变量获取参数，所以效率低

### SpringMVC实现RESTful服务

1. 导入jackson2包
2. 开启注解驱动<mvc:annotation-driven/>
3. Json交互=@RequestBody @ResponseBody

### SpringMVC如何返回json数据

在处理方法前加上@ResponseBody注解 或者 在控制器上使用@RestController

### mybatis中#{}与${}的区别

#{}解析的是占位符，可以防止SQL注入；参数值带有单引号

${}不能防止SQL注入；参数值不带单引号

### SpringBoot配置文件有哪几种类型？他们有什么区别？

1. SpringBoot的核心配置文件：application和bootstrap
2. 区别：

**Spring Cloud 的官方文档**

Spring Cloud构建于Spring Boot之上，在Spring Boot中有两种上下文，一种是bootstrap，另外一种是application，bootstrap是应用程序的父上下文，也就是说bootstrap加载优先于application。Bootstrap主要用于从额外的资源来加载配置信息，还可以在本地外部配置文件中解密属性。这两个上下文公用一个环境，它是任何Spring应用程序的外部属性的来源。Bootstrap里面的属性会优先加载，它们默认也不能被本地相同配置覆盖。

**对比application配置文件**

Bootstrap由父ApplicationContext加载，比application优先加载

Bootstrap里面的属性不能被覆盖

**Bootstrap/application的应用场景**

application用于SpringBoot自动化配置文件

bootstrap：

使用Spring Cloud Config注册中心时需要在bootstrap配置文件中添加链接到配置中心的配置属性来加载外部配置中心的配置信息

一些固定的不能被覆盖的属性

一些加密/解密的场景



1. 格式：.properties和.yml；.yml采取缩进格式，不支持@PropertySource注解导入配置

### 什么是死锁？怎么防止死锁