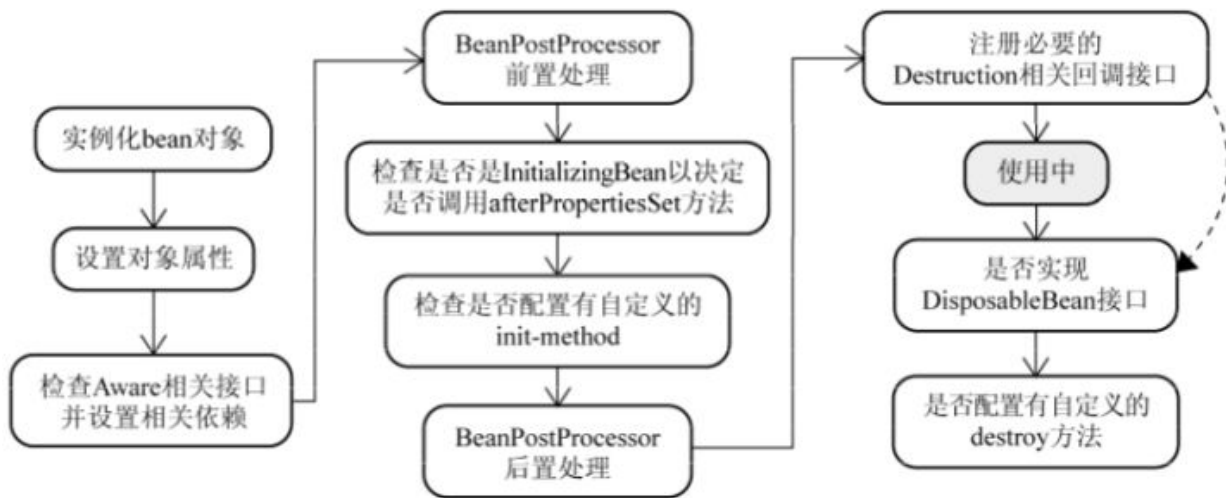


Spring 中的 bean 生命周期?



Bean 容器找到配置文件中 Spring Bean 的定义。

Bean 容器利用 Java Reflection API 创建一个Bean的实例。

如果涉及到一些属性值 利用 set() 方法设置一些属性值。

如果 Bean 实现了 BeanNameAware 接口，调用 setBeanName() 方法，传入Bean的名字。

如果 Bean 实现了 BeanClassLoaderAware 接口，调用 setBeanClassLoader() 方法，传入 ClassLoader 对象的实例。

与上面的类似，如果实现了其他 *.Aware 接口，就调用相应的方法。

如果有和加载这个 Bean 的 Spring 容器相关的 BeanPostProcessor 对象，执行 postProcessBeforeInitialization() 方法

如果Bean实现了 InitializingBean 接口，执行 afterPropertiesSet() 方法。

如果 Bean 在配置文件中的定义包含 init-method 属性，执行指定的方法。

如果有和加载这个 Bean的 Spring 容器相关的 BeanPostProcessor 对象，执行 postProcessAfterInitialization() 方法

当要销毁 Bean 的时候，如果 Bean 实现了 DisposableBean 接口，执行 destroy() 方法。

当要销毁 Bean 的时候，如果 Bean 在配置文件中的定义包含 destroy-method 属性，执行指定的方法。

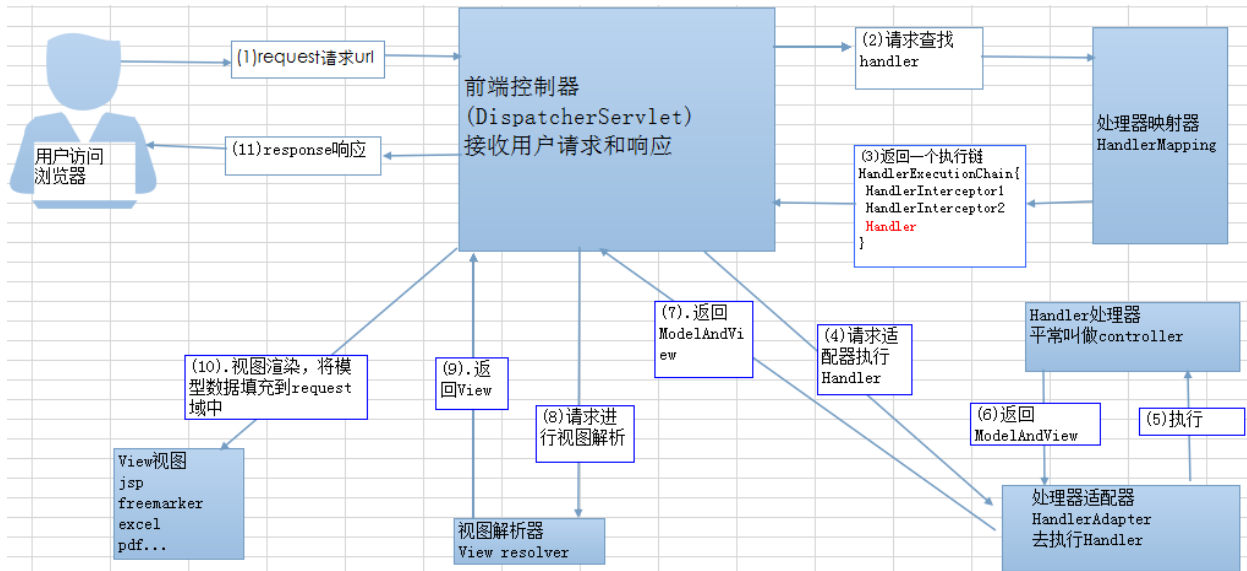
SpringMVC

MVC 是一种设计模式, Spring MVC 是一款很优秀的 MVC 框架。Spring MVC 可以帮助我们进行

更简洁的Web层的开发，并且它天生与 Spring 框架集成。Spring MVC 下我们一般把后端项目分

为 Service层（处理业务）、Dao层（数据库操作）、Entity层（实体类）、Controller层（控制

层，返回数据给前台页面）。



1. 客户端（浏览器）发送请求，直接请求到 DispatcherServlet 。
2. DispatcherServlet 根据请求信息调用 HandlerMapping ，解析请求对应的 Handler 。
3. 解析到对应的 Handler （也就是我们平常说的 Controller 控制器）后，开始由 HandlerAdapter 适配器处理。
4. HandlerAdapter 会根据 Handler 来调用真正的处理器开处理请求，并处理相应的业务逻辑。
5. 处理器处理完业务后，会返回一个 ModelAndView 对象， Model 是返回的数据对象，View 是个逻辑上的 View 。
6. ViewResolver 会根据逻辑 View 查找实际的 View 。
7. DispatserServlet 把返回的 Model 传给 View （视图渲染）。
8. 把 View 返回给请求者（浏览器）