

玩转Java注释:自动调用监听器Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/644/2021\\_2022\\_\\_E7\\_8E\\_A9\\_E8\\_BD\\_ACJava\\_c104\\_644818.htm](https://www.100test.com/kao_ti2020/644/2021_2022__E7_8E_A9_E8_BD_ACJava_c104_644818.htm) Listener，直译为侦听器或监听器，在面向对象的开发中经常需要用到。如果你需要启动或者停止基于Java的 Web系统中不同部分的代码，那么你可以使用一个简单的ServletContentListener来监听容器（container）里面的启动事件和停止事件。这个监听器可以使用java.util.ServiceLoader去寻找这些被侦听事件所对应的已注册类。这个方法不错，但是如果添加一个编译时间注释处理器会不会更好呢？如果你用

@Lifecycle(LifecycleEvent.STARTUP)注释一个静态方法，它将会在开机的时候被调用（在关机的时候被关掉）。处理器会产生类，并为了ServiceLoader而注册它们。你也可以把同样的机制用在任何事件总线（event-bus）模型上：在编译的时候注册 listener，并且带有注释，当事件被触发的时候总线会自动调用他们。从本质上讲，你可以通过注释实现用ServiceLoader在运行时间自动发现代码。实际过程中，其概念如下：1. 你用@EventListener对方法进行注释（可能包含一些元信息（meta-info））。2. 注释处理器为每个@EventListener方法生成一个EventDispatcher，包括注释中元信息需要的过滤器。3. 100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)