

Guice与Spring框架的区别 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/213/2021_2022_Guice_E4_B8_8ESp_c104_213739.htm 依赖注入，DI（Dependency Injection

），它的作用自然不必多说，提及DI容器，例如spring，picoContainer，EJB容器等等，近日，google诞生了更轻巧的DI容器.....Guice!废话不多讲了，先看看Guice是如何实现注入的吧。定义一个简单的service接口和它的实现吧：
public interface MyService ...{ void myMethod().}public class

MyServiceImpl implements MyService ...{ public void myMethod() ...{ System.out.println("Hello,World!"). }}以上是最普通的接口和其实现，没什么可说的。定义一个测试类,这个类里边包

括service对象的一个引用，这个对象是需要Guice进行注入的
import com.google.inject.Inject.public class Client ...{ MyService service. @Inject //告诉容器，这里的service对象的引用,需要进行注入 void setService(MyService service) ...{ //这里的方法名字可以任意定义 this.service = service. } public void myMethod() ...{ service.myMethod(). }} 这里除了加了一个@Inject,和Spring的配置没有任何的区别，@Inject，是表示对容器说，这里的service需要注射，等到运行的时候，容器会拿来一个实例给service，完成注射的过程。定义Guice的Module文件 告诉容器如何进行

注入
import com.google.inject.Binder.import com.google.inject.Module.import com.google.inject.Scopes.public class MyModule implements Module ...{ public void configure(Binder binder) ...{ binder.bind(MyService.class).to(MyServiceImpl.class).in(Scopes.SI

NGLETON). // 这句代码的意思是说：运行时动态的将MyServiceImpl对象赋给MyService定义的对象，而且这个对象是单例的。}}创建测试类

```
import com.google.inject.Guice; import com.google.inject.Injector; public class Test ...{ public static void main(String[] args) ...{ MyModule module = new MyModule(); // 定义注射规则 Injector injector = Guice.createInjector(module); // 根据注射规则，生成注射者 Client client = new Client(); injector.injectMembers(client); // 注射者将需要注射的bean,按照规则,把client这个客户端进行注射 client.myMethod(); }}
```

运行测试类，控制台输出：Hello,World! 完成注入过程下面看看Guice还有哪些其它的使用特性。1，如果在实现你确定MyService定义的对象，就要被注射为MyServiceImpl而不是其它的实现类的话，可以在MyService接口加上@ImplementedBy(MyServiceImpl.class)

```
import com.google.inject.ImplementedBy; @ImplementedBy(MyServiceImpl.class)
```

//我总觉得这样有点背离了依赖注入的初衷了#61514. 例如：spring的AspectJ，xdoclet等都是如此。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