



EBook Gratis

APRENDIZAJE

guice

Free unaffiliated eBook created from
Stack Overflow contributors.

#guice

Tabla de contenido

Acerca de	1
Capítulo 1: Empezando con guice	2
Observaciones.....	2
Examples.....	2
Configuración de un '¡Hola mundo!' ejemplo.....	2
Clases de muestra	2
Correr con gradle	3
Correr con maven	4
Capítulo 2: Método de interceptor Guice	6
Examples.....	6
Ejemplo simple.....	6
Capítulo 3: Usando Guice con Java WebSockets (JSR-356)	8
Examples.....	8
Configuración para obtener la inyección de Guice en los puntos finales de WebSocket.....	8
Capítulo 4: Usando Guice con Jersey 2 (JAX-RS RI)	9
Examples.....	9
Obtención de inyecciones de Guice en recursos JAX-RS.....	9
Creditos	10

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [guice](#)

It is an unofficial and free guice ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official guice.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con guice

Observaciones

Esta sección proporciona una descripción general de qué es una guía y por qué un desarrollador puede querer usarla.

También debe mencionar cualquier tema grande dentro de la guía, y vincular a los temas relacionados. Dado que la Documentación para Guice es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Configuración de un '¡Hola mundo!' ejemplo

Guice es una biblioteca de Java. Para usarlo, debe agregar un [archivo JAR](#) en la ruta de clase de su proyecto Java.

Clases de muestra

A continuación hay varias clases para un "¡Hola mundo!" ejemplo.

Una interfaz de un "servicio" hola:

```
public interface HelloWorldService {
    public void sayHello();
}
```

La implementación del servicio:

```
public class HelloWorldServiceImpl implements HelloWorldService {
    @Override
    public void sayHello() {
        System.out.println("Hello, world!");
    }
}
```

Un módulo de nombres de Guice. Es necesario para indicar a Guice que `HelloWorldServiceImpl` se inyectará donde se necesite un servicio de saludo.

```
import com.google.inject.AbstractModule;

public class HelloWorldModule extends AbstractModule {
    protected void configure() {
        bind(HelloWorldService.class).to(HelloWorldServiceImpl.class);
    }
}
```

Una clase principal donde se lleva a cabo la inyección real de un servicio de saludo:

```
import javax.inject.Inject;

import com.google.inject.Guice;
import com.google.inject.Injector;
import com.google.inject.Module;

public class Main {

    @Inject
    private HelloWorldService service;//hello service

    public static void main(String[] args) {

        Main main = new Main();

        Module module = new HelloWorldModule();
        Injector injector = Guice.createInjector(module);
        injector.injectMembers(main);//injects the implementation of the service

        main.testGuice();
    }

    public void testGuice()
    {
        service.sayHello();//usage of the service
    }
}
```

Correr con gradle

Para una rápida configuración y ejecución con Gradle 2.2. + Y Java 8:

1. Instala Gradle si aún no está instalado
2. Cree un directorio vacío y navegue en él con un shell habilitado para gradle
3. Crea un proyecto java vacío:

```
gradle init --type java-library
```

4. En el `build.gradle` generado automáticamente:

- **cambiar** `apply plugin: 'java'` para `apply plugin: 'application'`
- **agrega la siguiente línea**

```
mainClassName = 'Main'
```

- **en la sección de dependencias agregue una dependencia a una versión de guice, por ejemplo:**

```
dependencies {
```

```
...
compile group: 'com.google.inject', name: 'guice', version: '4.1.0'
...
}
```

5. Agregue las clases que se muestran arriba en el paquete predeterminado en `src/main/java`, cada uno en su propio archivo

6. Corre y disfruta

```
..> gradlew run
:compileJava
:processResources UP-TO-DATE
:classes
:run
Hello, world!

BUILD SUCCESSFUL

Total time: 3.595 secs
```

Correr con maven

Para una rápida configuración y ejecución con Maven 3+ y Java 8:

1. Instalar maven si aún no está instalado
2. Cree un directorio vacío y navegue en él con un shell habilitado para maven
3. Crea un proyecto java vacío:

```
mvn archetype:generate -DgroupId=com.example -DartifactId=guice -
DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

4. Cambiar al subdirectorio `guice`
5. En el `pom.xml` generado automáticamente:

- en el elemento `dependencies` agregue una dependencia a guice:

```
<dependency>
  <groupId>com.google.inject</groupId>
  <artifactId>guice</artifactId>
  <version>4.1.0</version>
</dependency>
```

- agrega el siguiente complemento a tu proyecto (permite una ejecución de prueba fácil)

```
<project>
  .....
  <build>
    <plugins>
```

```
    <plugin>
      <groupId>org.codehaus.mojo</groupId>
      <artifactId>exec-maven-plugin</artifactId>
      <version>1.5.0</version>
      <configuration>
        <mainClass>Main</mainClass>
      </configuration>
    </plugin>
  </plugins>
</build>
</project>
```

6. Agregue las clases que se muestran arriba en el paquete predeterminado en `src/main/java`, cada uno en su propio archivo

7. Corre y disfruta

```
...\guice>mvn exec:java
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building guice 1.0-SNAPSHOT
[INFO] -----
[INFO] --- exec-maven-plugin:1.5.0:java (default-cli) @ guice ---
Hello, world!
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.800 s
[INFO] Finished at: 2016-10-09T11:44:41+03:00
[INFO] Final Memory: 10M/309M
[INFO] -----
```

Lea Empezando con guice en línea: <https://riptutorial.com/es/guice/topic/3449/empezando-con-guice>

Capítulo 2: Método de interceptor Guice

Examples

Ejemplo simple

Método interceptado:

```
public class ExampleService implements Example {
    @MyAnnotation
    public doHomework() {
        System.out.println("working");
    }
}
```

Anotación:

```
@Retention(RetentionPolicy.RUNTIME) @Target (ElementType.METHOD)
public @interface MyAnnotation {}
```

Interceptor:

```
public class MyInterceptor implements MethodInterceptor {

    @Override
    public Object invoke(MethodInvocation arg0) throws Throwable {
        System.out.println("***** intercepted *****");
        return arg0.proceed();
    }
}
```

Módulo:

```
public class MyModule extends AbstractModule {

    @Override
    protected void configure() {
        bind(CallerService.class).to(Caller.class);
        bind(ExampleService.class).to(Example.class);

        bindInterceptor(Matchers.any(), Matchers.annotatedWith(MyAnnotation.class), new
MyInterceptor());
    }
}
```

Llamador:

```
public class CallerService implements Caller {
    @Inject
    private Client client;
```



```
public void call() {  
    client.doHomework();  
}  
}
```

Salida:

```
***** intercepted *****  
working
```

Lea Método de interceptor Guice en línea: <https://riptutorial.com/es/guice/topic/7907/metodo-de-interceptor-guice>

Capítulo 3: Usando Guice con Java WebSockets (JSR-356)

Examples

Configuración para obtener la inyección de Guice en los puntos finales de WebSocket

Primero, necesitamos un generador de puntos finales personalizado.

```
public class WSConfigurator extends ServerEndpointConfig.Configurator {
    @Inject
    private static Injector injector;

    @Override
    public <T> T getEndpointInstance(Class<T> endpointClass)
        throws InstantiationException
    {
        return injector.getInstance(endpointClass);
    }
}
```

Necesitamos arrancar el inyector en el configurador anterior desde uno de nuestros módulos Guice.

```
public class WebSocketModule extends AbstractModule {
    @Override
    protected void configure() {
        requestStaticInjection(WSConfigurator.class);
    }
}
```

Finalmente, podemos usar `@Inject` en el constructor de puntos finales.

```
@ServerEndpoint(
    value = "/ws/sync",

    configurator = WSConfigurator.class)
public class WSSync extends AsyncWebSocketServer {
    @Inject
    public WSSync(EventBus eventBus) {
        ...
    }
}
```

Lea [Usando Guice con Java WebSockets \(JSR-356\)](https://riptutorial.com/es/guice/topic/6462/usando-guice-con-java-websockets--jsr-356-) en línea:

<https://riptutorial.com/es/guice/topic/6462/usando-guice-con-java-websockets--jsr-356->

Capítulo 4: Usando Guice con Jersey 2 (JAX-RS RI)

Examples

Obtención de inyecciones de Guice en recursos JAX-RS

Necesitará el [puente guice](#) en su proyecto.

```
@ApplicationPath("api")
public class ApiRest extends ResourceConfig {
    @Inject
    public ApiRest(ServiceLocator serviceLocator, ServletContext servletContext) {
        packages("net.sargue.app.api");

        GuiceBridge.getGuiceBridge().initializeGuiceBridge(serviceLocator);
        GuiceIntoHK2Bridge guiceBridge = serviceLocator.getService(GuiceIntoHK2Bridge.class);
        Injector injector = (Injector) servletContext.getAttribute(Injector.class.getName());
        if (injector == null)
            throw new RuntimeException("Guice Injector not found");
        guiceBridge.bridgeGuiceInjector(injector);

        register(RolesAllowedDynamicFeature.class);
    }
}
```

Lea [Usando Guice con Jersey 2 \(JAX-RS RI\)](#) en línea:

<https://riptutorial.com/es/guice/topic/6376/usando-guice-con-jersey-2--jax-rs-ri>

Creditos

S. No	Capítulos	Contributors
1	Empezando con guice	Community , Lachezar Balev
2	Método de interceptor Guice	David
3	Usando Guice con Java WebSockets (JSR-356)	sargue
4	Usando Guice con Jersey 2 (JAX-RS RI)	sargue