Hibernate 5 - Event listener example

In last post, we learned how to [configure the interceptor](https://www.boraji.com/hibernate-5-interceptor-example) in Hibernate application. Interceptors in Hibernate are primarily used to inspect the object properties before some events.

Interceptor’s methods such as onSave(), onLoad() etc. are invoked by Hibernate default event listeners. For example, the Interceptor#onLoad() is called by the DefaultPreLoadEventListener.

Hibernate native event system is more flexible then interceptor. It allows us to implement custom event listeners to listen the PreLoadEvent, PostLoadEvent, PostInsertEvent and [more](http://docs.jboss.org/hibernate/orm/current/javadocs/org/hibernate/event/spi/package-summary.html).

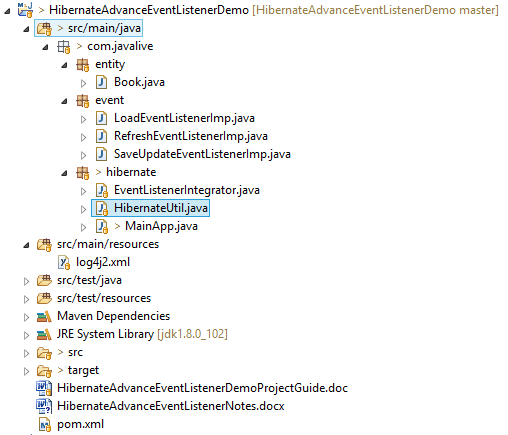
Let’s see an example how to implement the event listener in Hibernate application.

Tools and technologies used for this application are -

* Hibernate ORM 5.2.12.Final
* Log4j 2.8.2
* JavaSE 1.8
* MySQL Server 5.7.12
* Eclipse Oxygen.1
* Maven 3.3.9

### Project structure

Final project structure of our application will look like as follows.



### Jar dependencies

Add the following jar dependencies in pom.xml file of your maven project.

**pom.xml**

<project xmlns="<http://maven.apache.org/POM/4.0.0>" xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"

xsi:schemaLocation="<http://maven.apache.org/POM/4.0.0> <http://maven.apache.org/xsd/maven-4.0.0.xsd>">

<modelVersion>4.0.0</modelVersion>

<groupId>com.boraji.tutorial.hibernate</groupId>

<artifactId>hibernate-event-example</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>hibernate-event-example</name>

<dependencies>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.2.12.Final</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>6.0.6</version>

</dependency>

<dependency>

<groupId>org.apache.logging.log4j</groupId>

<artifactId>log4j-core</artifactId>

<version>2.8.2</version>

</dependency>

<dependency>

<groupId>org.apache.logging.log4j</groupId>

<artifactId>log4j-api</artifactId>

<version>2.8.2</version>

</dependency>

</dependencies>

<build>

<sourceDirectory>src/main/java</sourceDirectory>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.5.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

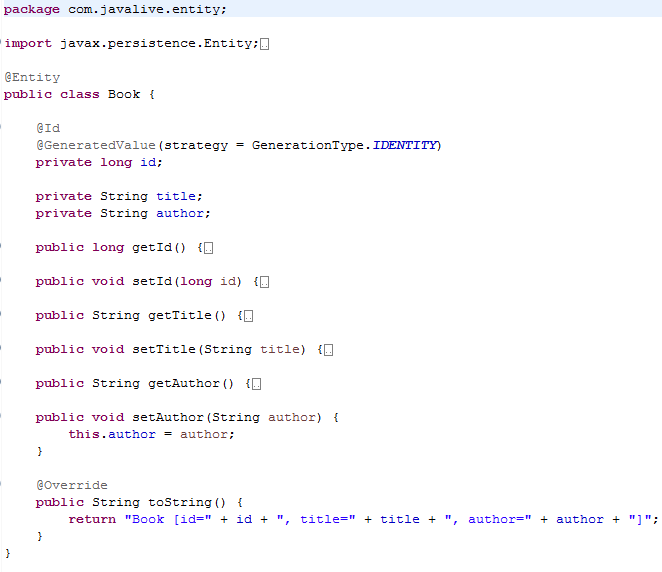
</plugins>

</build>

</project>

### Entity class

Create a simple @Entity class as follows.

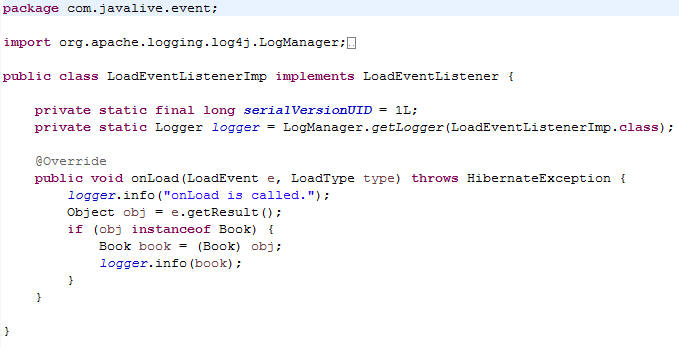


### Implementing event listeners

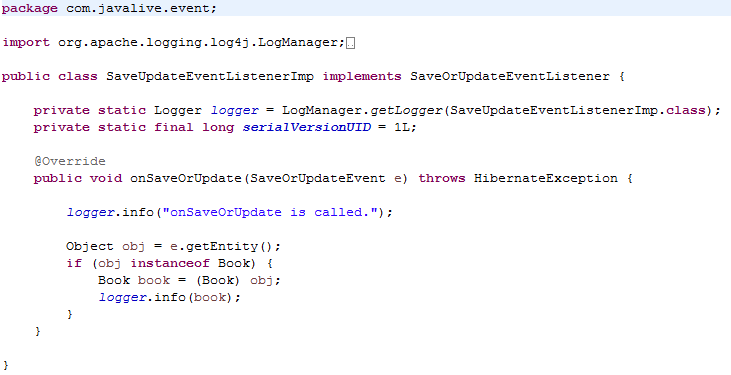
In Hibernate, all event listener interfaces are available in org.hibernate.event.spi package. You can use any event listener based on the org.hibernate.event.spi.EventType.

In this example, I will show you how to implement the LoadEventListener, SaveOrUpdateEventListener and RefreshEventListener as follows.

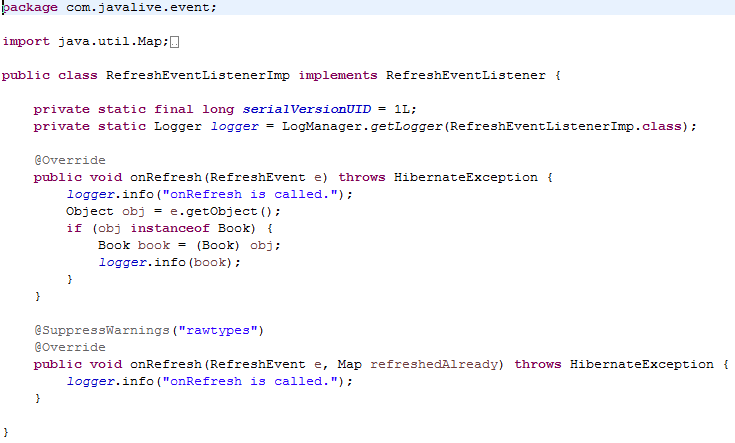
**LoadEventListenerImp.java**



**SaveUpdateEventListenerImp.java**



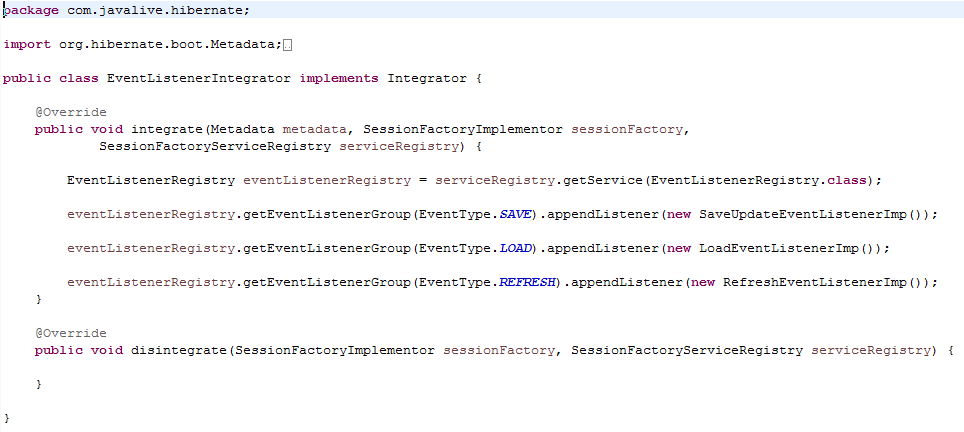
**RefreshEventListenerImp.java**



### Registering event listeners

To register event listeners, you need to create an integrator class by implement the org.hibernate.integrator.spi.Integrator interface as follows.

**EventListenerIntegrator.java**



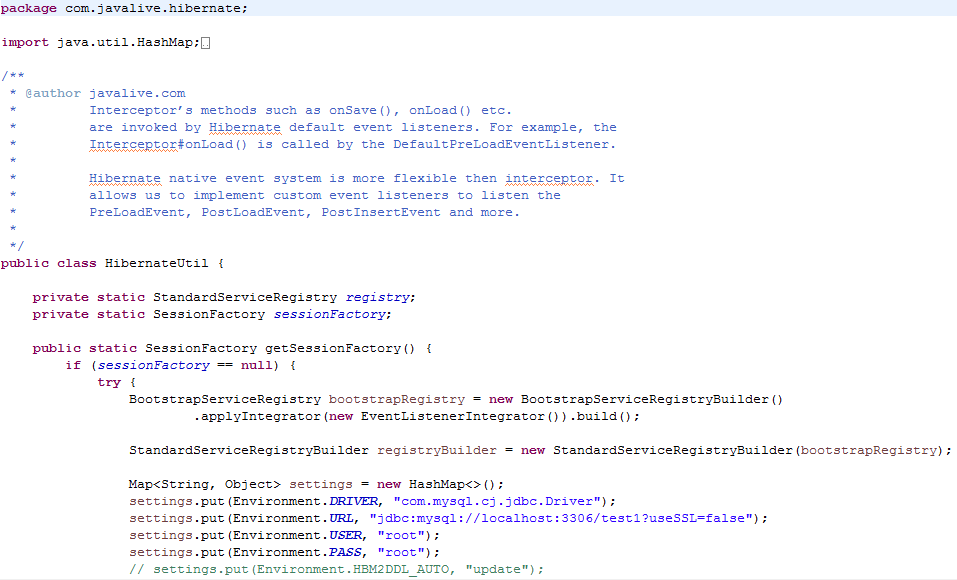
### Hibernate Utility class

Create a helper class HibernateUtil to bootstrap hibernate.

Add the EventListenerIntegrator using the BootstrapServiceRegistryBuilder#applyIntegrator() method to be applied to the bootstrap registry.

Map the Book entity using the #MetadataSources.addAnnotatedClass() method.

**HibernateUtil.java**





### Logging event listeners

To enable logging, create a log4j2.xml file under src/main/resources folder and write the following code in it.

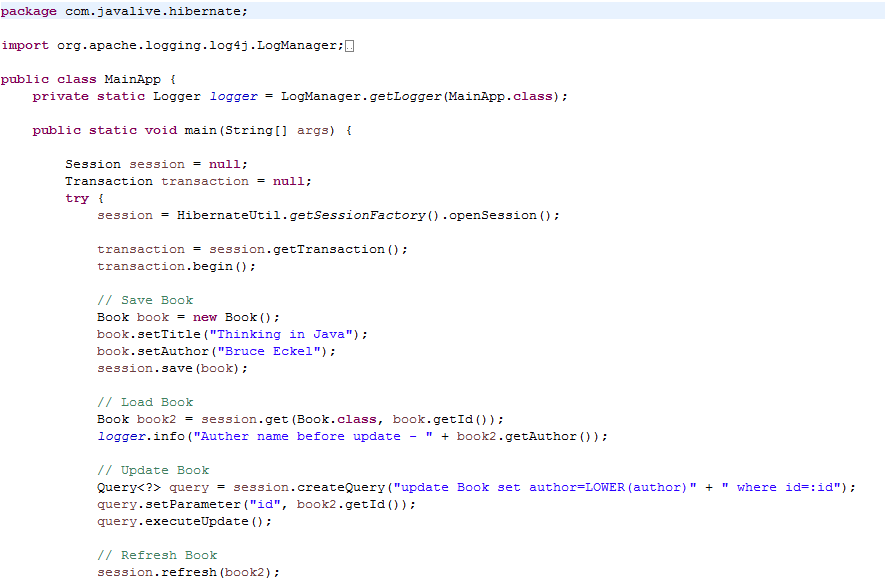
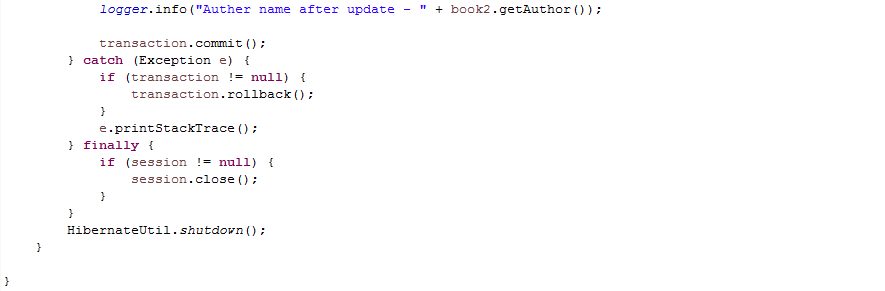
**log4j2.xml**



### Run Application

Create main class to test the registered event listeners.

**MainApp.java**

After executing the MainApp, the output of your program will look like as follows.

