Hibernate 5 - datasource-proxy configuration example

**Technologies used:** Java SE 1.8 | Hibernate 5.2.10.Final | datasource-proxy 1.4.1 | Maven 3.3.9 | MySQL 5.7.12 | Eclipse Neon.3

We can use **[datasource-proxy](https://github.com/ttddyy/datasource-proxy" \t "_blank)** API with Hibernate application for query and parameter logging.  By using this API, you can log query execution time, parameters, number of select/update/delete/insert queries, type of sql statement (Prepared or Statement), batch size etc.

The datasource-proxy API also allows us to performing certain tasks before/after query execution.

Let’s write a simple hibernate application to see how we can log sql queries and parameters with datasource-proxy.

Jar Dependencies

In pom.xml file of your maven project, add the dependencies below.

<dependencies>

*<!-- Mysql Connector -->*

<dependency>

<groupId>mysql</groupId>

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

<version>6.0.5</version>

</dependency>

*<!-- Hibernate 5.2.9 Final -->*

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.2.10.Final</version>

</dependency>

*<!-- datasource-proxy -->*

<dependency>

<groupId>net.ttddyy</groupId>

<artifactId>datasource-proxy</artifactId>

<version>1.4.1</version>

</dependency>

*<!-- commons-logging for query metrics. -->*

<dependency>

<groupId>commons-logging</groupId>

<artifactId>commons-logging</artifactId>

<version>1.2</version>

</dependency>

</dependencies>

Hibernate Utility class + ProxyDataSource

To integrate datasource-proxy with Hibernate, you need to create a ProxyDataSource object.

ProxyDataSource dataSource = ProxyDataSourceBuilder.create(actualDataSource)

.logQueryByCommons(CommonsLogLevel.INFO)

.countQuery()

.multiline()

.build();

Then, set the ProxyDataSource object in hibernate setting values.

Map<String, Object> settings = **new** HashMap<>();

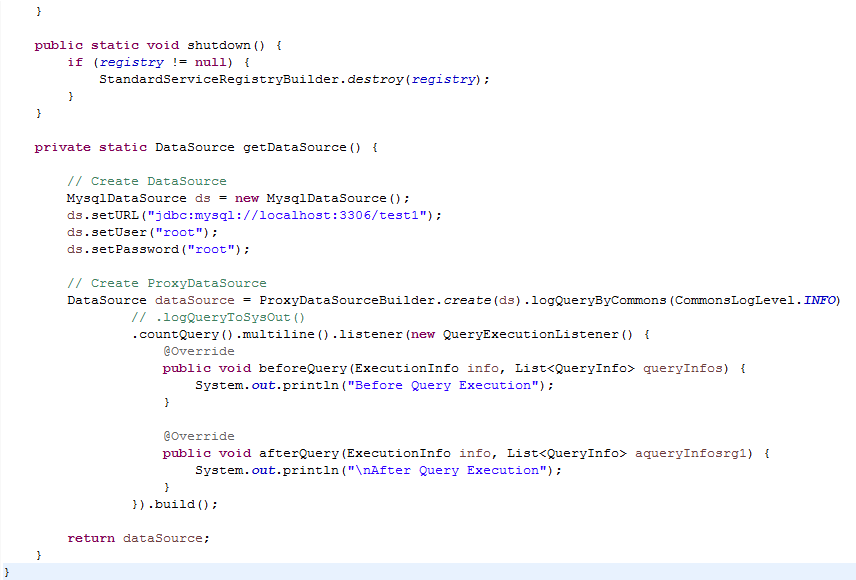
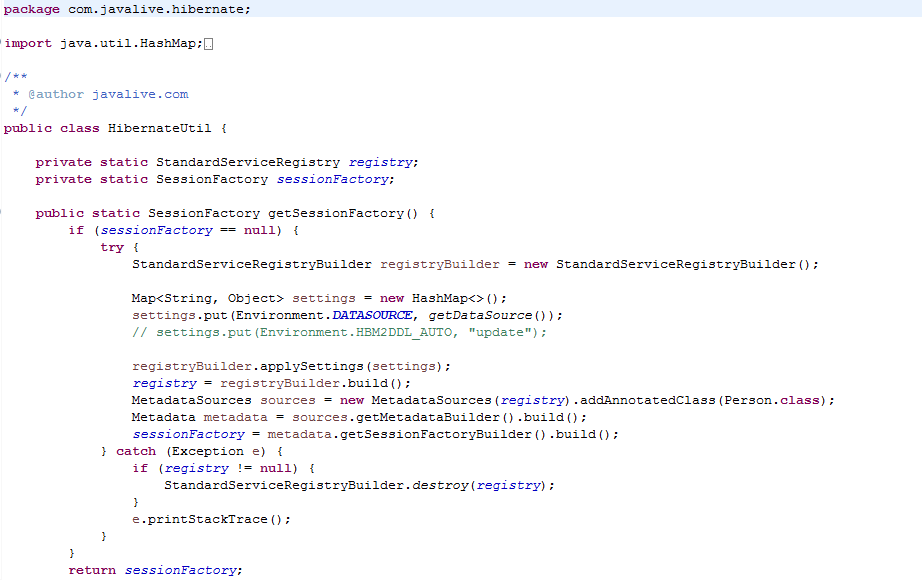
settings.put(Environment.DATASOURCE, getDataSource());

*//...*

registryBuilder.applySettings(settings);

Here is the complete example of HibernateUitl helper class to bootstrap application.

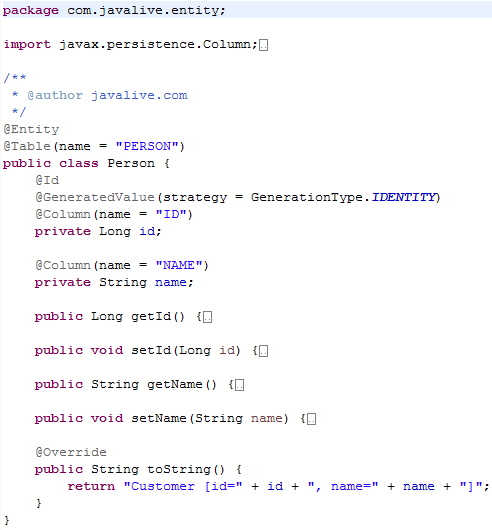
**HibernateUtil.java**

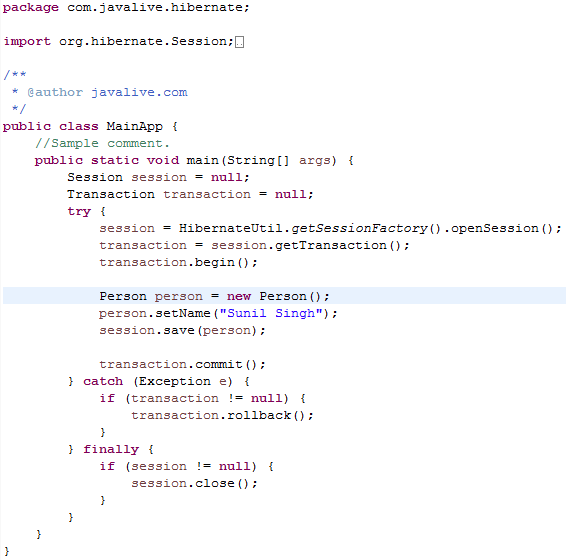


Entity class

A simple model class with JPA annotations.

**Person.java**





**Output**

