Hibernate-Logging Workshop

Recap:

In the previous workshops, we have created a project called HibernateProject.
In this project, we have a Student class. By using Hibernate, we could create a student table in our database, and we could manipulate the student table from Java (adding, deleting, retrieving and updating data).
We could even change the name of the table and the variables.
We have learned how to use field access and property access.

In this workshop:

- We continue with the same project.
- We will set up our logging process: we will add log4j to our project in order to improve our logging.

Logging

Here for the logging we use *slf4j* as the abstraction and *log4j* as the framework.
We already have the slf4j jar file in the library of the project. So we do not need to add it.
But we need to add log4j.
You are provided with the *log4j.properties* *here*.
Drag and drop or copy the *log4j.properties* file in your **resources** directory.

And there are dependencies for the following which needs to be added in your pom file:

log4j-1.2.17.jar
slf4j-log4j12-1.7.0.jar

```
<!-- https://mvnrepository.com/artifact/log4j/log4j -->
<dependency>
   <groupId>log4j</groupId>
   <artifactId>log4j</artifactId>
   <version>1.2.17</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.slf4j/slf4j-log4j12 -->
<dependency>
   <groupId>org.slf4j</groupId>
   <artifactId>slf4j-log4j12</artifactId>
   <version>1.7.0</version>
   <scope>test</scope>
</dependency>
```

Open the *log4j.properties* file.
There are two lines in the file that specify how much information we want to get.

```
log4j.logger.org.hibernate=warn

log4j.logger.org.hibernate.type=info
```

The first line says that it shows only the warnings.
We can choose *info* or *debug* instead of *warn*. But *warn* is much cleaner than the two other ones.

The second line is for sql running.
we have a PreparedStatement running behind and we cannot see the values. instead we see (?,?,?,?).
If we want to get more information and also the values for them we can change *info* to *trace*

Open hibernate.cfg.xml

There is a line:

```
<property name="show_sql">true</property>
```

This line enables the sql log. But if we set it to false, we would not see the sql log anymore.
Under this line we can add:

```
<property name="format_sql">true</property>
<property name="use_sql_comments">true</property>
```

The first line splits the sql to separate lines and the second one adds comments.

Run the code and see the console to see the tailored log.