

Task 1: As explained in the lecture on Hello World with Hibernate and JPA Annotations, create a `hello-world` database and a `message` table in it.

Task 2: Create a `Message` class and map it to the `message` table using hibernate mapping file (XML-based metadata), then create an object of the `Message` object persist it to the `message` table.

Task 3: Create a `Message` entity class using JPA annotations and map it to the `message` table using JPA annotations (annotation-based metadata), then create a `Message` entity object and persist its state to the `message` table.

***The source code files for the lecture on "Hello World with Hibernate and JPA Annotations" are available to be downloaded with this lab exercise. You could use them to complete the given tasks successfully.

You could also look at the source code inside all the downloadable files below.

Source Code: [Hello World with Hibernate and JPA Annotations](#)

Schema for `hello-world` database

```
CREATE DATABASE `hello-world`;
USE `hello-world`;
CREATE TABLE `message` (
  `ID` BIGINT(20) NOT NULL AUTO_INCREMENT,
  `TEXT` VARCHAR(255) NULL DEFAULT NULL,
  PRIMARY KEY (`ID`)
);
```

HibernateUtil.java

```
package util;

import org.hibernate.SessionFactory;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;

public class HibernateUtil {

    private static final SessionFactory sessionFactory = buildSessionFactory();

    private static SessionFactory buildSessionFactory() {
        try {
            // Create the SessionFactory from hibernate.cfg.xml
            Configuration configuration = new Configuration().configure("hibernate.cfg.xml");
            return configuration.buildSessionFactory( new
StandardServiceRegistryBuilder().applySettings( configuration.getProperties() ).build() );
        }
        catch (Throwable ex) {
            // Make sure you log the exception, as it might be swallowed
            System.err.println("Initial SessionFactory creation failed." + ex);
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
}
```

```
}  
  
}
```

hibernate.cfg.xml

(to be placed in the classpath of your application)

```
<?xml version='1.0' encoding='utf-8'?>  
<!DOCTYPE hibernate-configuration PUBLIC  
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"  
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">  
  
<hibernate-configuration>  
    <session-factory>  
  
        <!-- Database connection settings -->  
        <property name="connection.driver_class">com.mysql.jdbc.Driver</property>  
        <property name="connection.url">jdbc:mysql://localhost:3306/hello-world</property>  
        <property name="connection.username">root</property>  
        <property name="connection.password">password</property>  
  
        <!-- SQL dialect -->  
        <property name="dialect">org.hibernate.dialect.MySQLDialect</property>  
  
        <!-- Echo all executed SQL to stdout -->  
        <property name="show_sql">true</property>  
  
        <!-- Use XML-based mapping metadata -->  
        <!-- <mapping resource="domain/Message.hbm.xml"/> -->  
  
        <!-- Use Annotation-based mapping metadata -->  
        <mapping class="entity.Message"/>  
  
    </session-factory>  
</hibernate-configuration>
```

Message.hbm.xml

(to be placed in the same package as the Message.java : "domain" package)

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping package="domain">
    <class name="Message" table="MESSAGE">
        <id name="id" column="ID">
            <generator class="native" />
        </id>
        <property name="text" type="string" column="TEXT"/>
    </class>
</hibernate-mapping>
```

HelloWorldClient.java

```
package client;

import org.hibernate.Session;

import util.HibernateUtil;
import entity.Message;

public class HelloWorldClient {
    public static void main(String[] args) {

        Session session = HibernateUtil.getSessionFactory().openSession();
        session.beginTransaction();

        Message message = new Message( "Hello World with Hibernate & JPA
Annotations" );

        session.save(message);

        session.getTransaction().commit();
        session.close();
    }
}
```

```
}  
}
```

Message.java

(using JPA annotations)

```
package entity;
```

```
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import javax.persistence.GenerationType;  
import javax.persistence.Id;  
import javax.persistence.Table;
```

```
@Entity  
@Table(name="message")  
public class Message {
```

```
    @Id  
    @GeneratedValue(strategy=GenerationType.AUTO)  
    @Column(name="ID")  
    private Long id;
```

```
    @Column(name="TEXT")  
    private String text;
```

```
    public Message() {}  
    public Message(String text) {  
        this.text = text;  
    }
```

```
    @Override  
    public String toString() {  
        return "Message [id=" + id + ", text=" + text + "];"  
    }
```

```
}
```

Message.java

(to be used with Hibernate mapping file, i.e. Message.hbm.xml)

```
package domain;

public class Message {
    private Long id;
    private String text;

    public Message() {}
    public Message(String text) {
        this.text = text;
    }

    public Long getId() {
        return id;
    }
    private void setId(Long id) {
        this.id = id;
    }
    public String getText() {
        return text;
    }
    public void setText(String text) {
        this.text = text;
    }

    @Override
    public String toString() {
        return "Message [id=" + id + ", text=" + text + "];"
    }
}
```