

Hibernate

Download Hibernate (in the section ORM): www.hibernate.org

In order to exemplify the concepts of Hibernate we will make some modifications in the project classspring.

Step 1: Download the file with the libraries for Hibernate. In the folder WEB-INF/lib, add all files in lib/required and the file in lib/jpa.

Step 2: Download the file persistence.xml to a folder named META-INF inside the folder src. **Note:** you have to create the folder.

Step 3: Download the file log4j.properties to the folder src.

Step 4: In the class Student.java, add the Hibernate annotations:

```
@Entity
@Table(name="students", schema="public")
public class Student {
    @Id
    @GeneratedValue
    private Long id;
    @Column(name="name")
    private String name;
    @Column(name="email")
    private String email;
    @Column(name="address")
    private String address;
    @Column(name="graduated")
    private boolean graduated;
    // getters and setters
}
```

Step 5: Modify the action insertStudent as follows:

```
@RequestMapping("insertStudent")
public String insert(Student student) {
    //StudentDAO dao = new StudentDAO();
    //dao.insert(student);

    EntityManagerFactory factory = Persistence
        .createEntityManagerFactory("student");
    EntityManager manager = factory.createEntityManager();

    manager.getTransaction().begin();
    manager.persist(student);
    manager.getTransaction().commit();
    manager.close();
    return "redirect:listStudents";
}
```

Step 6: Modify the action updateStudent as follows:

```
@RequestMapping("updateStudent")
public String update(Student student) {
    // StudentDAO dao = new StudentDAO();
    // dao.update(student);

    EntityManagerFactory factory = Persistence
        .createEntityManagerFactory("student");
    EntityManager manager = factory.createEntityManager();
    manager.getTransaction().begin();
    manager.merge(student);
    manager.getTransaction().commit();
    manager.close();
    return "redirect:listStudents";
}
```

Step 7: Modify the action removeStudent as follows:

```
@RequestMapping("removeStudent")
public String remove(Student student) {
    // StudentDAO dao = new StudentDAO();
    // dao.remove(student);
    EntityManagerFactory factory = Persistence
        .createEntityManagerFactory("student");
    EntityManager manager = factory.createEntityManager();
    manager.getTransaction().begin();
    Student found = manager.find(Student.class, student.getId());

    manager.remove(found);
    manager.getTransaction().commit();
    manager.close();
    return "redirect:listStudents";
}
```

Step 8: Modify the action showStudent as follows:

```
@RequestMapping("showStudent")
public String show(Long id, Model model) {
    EntityManagerFactory factory = Persistence
        .createEntityManagerFactory("student");
    EntityManager manager = factory.createEntityManager();

    Student student = manager.find(Student.class, id);
    manager.close();

    // StudentDAO dao = new StudentDAO();
    // model.addAttribute("student", dao.searchById(id));
    model.addAttribute("student", student);
    return "student/show";
}
```

Step 9: Modify the action listStudents as follows:

```
@RequestMapping("listStudents")
public String list(Model model) {
    // StudentDAO dao = new StudentDAO();
    // model.addAttribute("students", dao.getList());

    EntityManagerFactory factory = Persistence
        .createEntityManagerFactory("student");
    EntityManager manager = factory.createEntityManager();

    @SuppressWarnings("unchecked")
    List<Student> alunos = manager
        .createQuery("select a from Student as a").getResultList();
    model.addAttribute("students", students);

    manager.close();
    return "student/list";
}
```

Step 10: Modify the action changeStatus to use Hibernate/JPA as in the other operations.