

Hibernate

Download Hibernate (na seção ORM): www.hibernate.org

Para exemplificar os conceitos de Hibernate vamos fazer algumas modificações no projeto aulaspring.

Passo 1: Baixe o arquivo com as bibliotecas necessárias no site da disciplina na aula de Hibernate. Na pasta WEB-INF/lib, coloque todos os arquivos em lib/required e o arquivo em lib/jpa.

Passo 2: Baixe o arquivo persistence.xml e coloque-o em uma pasta chamada META-INF dentro da pasta src. **Observação:** é preciso criar a pasta META-INF.

Passo 3: Baixe o arquivo log4j.properties e coloque-o na pasta src.

Passo 4: Na classe Aluno.java, acrescente as anotações do Hibernate:

```
@Entity
@Table(name="alunos", schema="public")
public class Aluno {
    @Id
    @GeneratedValue
    private Long id;
    @Column(name="nome")
    private String nome;
    @Column(name="email")
    private String email;
    @Column(name="endereco")
    private String endereco;
    @Column(name="formado")
    private boolean formado;
    // getters and setters
}
```

Passo 5: Modifique a ação insereAluno como segue:

```
@RequestMapping("insereAluno")
public String insere(Aluno aluno) {
    //AlunoDAO dao = new AlunoDAO();
    //dao.insere(aluno);

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

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

Passo 6: Modifique a ação alteraAluno como segue:

```
@RequestMapping("alteraAluno")
public String altera(Aluno aluno) {
    // AlunoDAO dao = new AlunoDAO();
    // dao.altera(aluno);

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

Passo 7: Modifique a ação removeAluno como segue:

```
@RequestMapping("removeAluno")
public String remove(Aluno aluno) {
    // AlunoDAO dao = new AlunoDAO();
    // dao.remove(aluno);
    EntityManagerFactory factory = Persistence
        .createEntityManagerFactory("aluno");
    EntityManager manager = factory.createEntityManager();
    manager.getTransaction().begin();
    Aluno encontrado = manager.find(Aluno.class, aluno.getId());

    manager.remove(encontrado);
    manager.getTransaction().commit();
    manager.close();
    return "redirect:listaAlunos";
}
```

Passo 8: Modifique a ação exhibeAluno como segue:

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

    Aluno aluno = manager.find(Aluno.class, id);
    manager.close();

    // AlunoDAO dao = new AlunoDAO();
    // model.addAttribute("aluno", dao.buscaPorId(id));
    model.addAttribute("aluno", aluno);
    return "aluno/exibe";
}
```

Passo 9: Modifique a ação listaAlunos como segue:

```
@RequestMapping("listaAlunos")
public String lista(Model model) {
    // AlunoDAO dao = new AlunoDAO();
    // model.addAttribute("alunos", dao.getLista());

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

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

    manager.close();
    return "aluno/lista";
}
```

Passo 10: Modifique a ação mudaStatus para usar Hibernate/JPA como nas outras operações.