



Desenvolvimento de Software para WEB

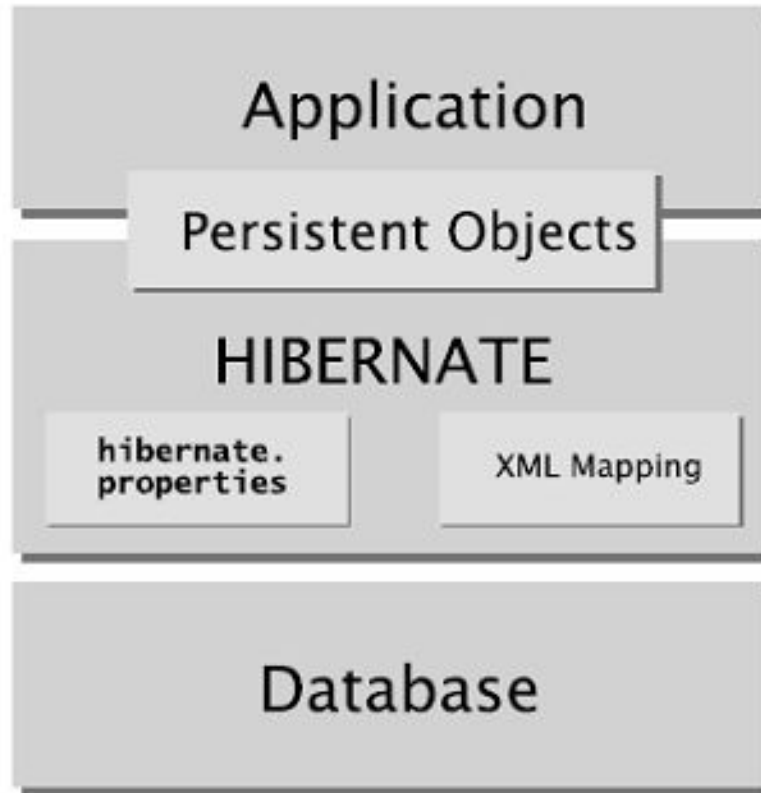
Aula 18 - JPA - Hibernate

Professor: Anderson Almada

Introdução

- Mapeamento objeto-relacional (ORM)
- O Hibernate é uma ferramenta ORM open source e líder de mercado (Java)
- Especificação Java Persistence API (JPA).
- O Hibernate abstrai o seu código SQL, toda a camada JDBC e o SQL será gerado em tempo de execução.
 - Possibilidade de trocar de banco de dados sem ter de alterar código Java, já que isso fica como responsabilidade da ferramenta.

Introdução



Instalação Hibernate

- <https://hibernate.org/orm/releases/>

```
<dependency>
```

```
  <groupId>org.hibernate</groupId>
```

```
  <artifactId>hibernate-agroal</artifactId>
```

```
  <version>5.3.10.Final</version>
```

```
  <type>pom</type>
```

```
</dependency>
```

Crie uma classe

- Tarefa
 - <https://pastebin.com/63DbYM49>

Adicione as devidas anotações

- Tarefa
 - @Entity
 - @Id
 - @GeneratedValue
 - @Column(columnDefinition="text")
 - <https://pastebin.com/w1RqCsQW>

Configurar o jpa

- src/main/resources/META-INF/persistence.xml
 - <https://pastebin.com/1k05Ng7j>
- Agora, basta utilizar a API
 - ```
EntityManagerFactory factory =
 Persistence.createEntityManagerFactory("tarefas");
```

# Crie uma classe de Test

---

```
public class GeraTabelas {
 public static void main(String[] args) {
 EntityManagerFactory factory = Persistence.
 createEntityManagerFactory("tarefas");

 factory.close();
 }
}
```

**Rode como um java application e verifique no pgadmin se foi criada a tabela !!**





Quick Access

Project Explorer

- Java Resources
  - src/main/java
    - rest
    - rest.controller
    - rest.dao
    - rest.model
    - rest.util
      - DbUtil.java
      - GerarTabela.java
  - src/main/resources
    - META-INF
      - persistence.xml
      - db.properties
      - uploads.properties
  - src/test/java
  - Libraries
- JavaScript Resources
- Deployed Resources
- src
  - main
    - java
  - resources
    - META-INF

Tarefa.java

GerarTabela.java

Web Browser

1 package rest.util;



Markers



Properties



Servers



Data Source Explorer



Snippets



Problems



Console

&lt;terminated&gt; GerarTabela [Java Application] /usr/lib/jvm/java-8-oracle/bin/java (19 de mai de 2019 17:24:46)

INFO: HHH10001005: using driver [org.postgresql.Driver] at URL [jdbc:postgresql://localhost:5432/projeto]

mai 19, 2019 5:24:47 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator

INFO: HHH10001001: Connection properties: {user=postgres, password=\*\*\*\*}

mai 19, 2019 5:24:47 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator

INFO: HHH10001003: Autocommit mode: false

mai 19, 2019 5:24:47 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl\$PooledConnections &lt;i

INFO: HHH000115: Hibernate connection pool size: 20 (min=1)

mai 19, 2019 5:24:48 PM org.hibernate.dialect.Dialect &lt;init&gt;

INFO: HHH000400: Using dialect: org.hibernate.dialect.PostgreSQLDialect

mai 19, 2019 5:24:48 PM org.hibernate.engine.jdbc.env.internal.LobCreatorBuilderImpl useContextualLobCreation

INFO: HHH000424: Disabling contextual LOB creation as createClob() method threw error : java.lang.reflect.InvocationTargetException

mai 19, 2019 5:24:48 PM org.hibernate.type.BasicTypeRegistry register

INFO: HHH000270: Type registration [java.util.UUID] overrides previous : org.hibernate.type.UUIDBinaryType@53941c2f

mai 19, 2019 5:24:49 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedCo

INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitial

Hibernate:

```
create table Tarefa (
 id int8 not null,
 descricao text,
 finalizado boolean not null,
 primary key (id)
)
```

mai 19, 2019 5:24:49 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl stop

INFO: HHH10001008: Cleaning up connection pool [jdbc:postgresql://localhost:5432/projeto]

# Trabalhando com os objetos

---

```
EntityManagerFactory factory =
Persistence.createEntityManagerFactory("tarefas");

EntityManager manager = factory.createEntityManager();

manager.close();
factory.close();
```

# Inserindo um objeto

---

```
public static void addTarefas(Tarefa tarefa) {
 EntityManager manager = factory.createEntityManager();

 manager.getTransaction().begin();
 manager.persist(tarefa);
 manager.getTransaction().commit();

 manager.close();
}
```

## Recuperando um objeto

---

```
public static Tarefa getTarefa(int i) {
 EntityManager manager = factory.createEntityManager();

 Tarefa encontrada = manager.find(Tarefa.class, i);

 manager.close();

 return encontrada;
}
```

# Recuperando todos os objetos

---

```
public static List<Tarefa> getTarefas() {
 EntityManager manager = factory.createEntityManager();

 List<Tarefa> encontradas = manager.createQuery("from Tarefa",
Tarefa.class).getResultList();

 manager.close();

 return encontradas;
}
```

## Recuperando um objeto com query

---

```
public static Tarefa getTarefaByQuery(int i) {
 EntityManager manager = factory.createEntityManager();

 Tarefa encontrada = manager.createQuery("from Tarefa where id = " +
i, Tarefa.class).getResultList().get(0);

 manager.close();

 return encontrada;
}
```

## Atualizando um objeto

---

```
public static void atualizaTarefa(Tarefa tarefa) {
 EntityManager manager = factory.createEntityManager();

 manager.getTransaction().begin();
 manager.merge(tarefa);
 manager.getTransaction().commit();

 manager.close();
}
```

Atualize os dados (menos o id)

## Remove um objeto

---

```
public static void removeTarefa(int i) {
 EntityManager manager = factory.createEntityManager();

 Tarefa encontrada = manager.find(Tarefa.class, i);

 manager.getTransaction().begin();
 manager.remove(encontrada);
 manager.getTransaction().commit();

 manager.close();
}
```



# TarefaDAO com Hibernate

---

- <https://pastebin.com/A6Zh1mFy>

# Links importantes

---

- <https://hibernate.org/>
- <https://www.caelum.com.br/apostila-java-web/uma-introducao-pratica-ao-jpa-com-hibernate/#java-persistence-api-e-frameworks-orm>



# Dúvidas??

E-mail: [almada@crateus.ufc.br](mailto:almada@crateus.ufc.br)