PAPW

Alberto Benavides

Ago - Dic 2018

17. Flujo de BD

1. Añadir dependencias a `pow.xml`

- https://mvnrepository.com/artifact/mysql/mysql-connector-java
- https://mvnrepository.com/artifact/org.springframework/spring-jdbc
- https://mvnrepository.com/artifact/org.springframework/spring-tx

2. Crear tabla

```
CREATE TABLE User(
   ID   INT NOT NULL AUTO_INCREMENT,
   NAME VARCHAR(20) NOT NULL,
   AGE  INT NOT NULL,
   PRIMARY KEY (ID)
);
```

3. Crear DAO

- DAO: Data Access Object

```java
package com.example;
import java.util.List;
import javax.sql.DataSource;

public interface UserDAO {
   public void setDataSource(DataSource ds);

   public void create(String name, Integer age);

   public Student getUser(Integer id);

   public List<User> listUsers();

   public void delete(Integer id);

   public void update(Integer id, Integer age);
}
```

4. Crear modelo

```java
package com.example;
public class User {
  private Integer age;
  private String name;
  private Integer id;

  public void setAge(Integer age) {this.age = age;}
  public Integer getAge() {return age;}

  public void setName(String name) {this.name = name;}
  public String getName() {return name;}

  public void setId(Integer id) {this.id = id;}
  public Integer getId() {return id;}
}
```

5. Crear un mapeo

```java
package com.example;

import java.sql.ResultSet;
import java.sql.SQLException;
import org.springframework.jdbc.core.RowMapper;

public class UserMapper implements RowMapper<User> {
   public User mapRow(ResultSet rs, int rowNum)
     throws SQLException {
      User user = new User();
      user.setId(rs.getInt("id"));
      user.setName(rs.getString("name"));
      user.setAge(rs.getInt("age"));

      return user;
   }
}
```

6. Plantilla JDBC

```
package com.examplw;
import java.util.List;
import javax.sql.DataSource;
import org.springframework.jdbc.core.JdbcTemplate;
public class UserJDBCTemplate implements UserDAO {
   // Siguientes diapositivas
}
```

Funciones UserJDBCTemplate

```
private JdbcTemplate jdbcTemplateObject;

public void setDataSource(Connection connection) {
  this.jdbcTemplateObject =
    new JdbcTemplate(
      new SingleConnectionDataSource(connection, false)
    );
}

public void create(String name, Integer age) {
  String SQL =
    "insert into User (name, age) values (?, ?)";
  jdbcTemplateObject.update(SQL, name, age);
  System.out.println(
    "Registro creado = " + name + " Age = " + age
  );
  return;
}
```

Funciones UserJDBCTemplate

```
public User getUser(Integer id) {
  String SQL = "select * from User where id = ?";
  User user = jdbcTemplateObject.queryForObject(SQL,
     new Object[]{id}, new UserMapper());
  return user;
}

public List<User> listUsers() {
  String SQL = "select * from User";
  List <User> users =
    jdbcTemplateObject.query(SQL, new UserMapper());
  return users;
}
```

Funciones UserJDBCTemplate

```java
public void delete(Integer id) {
  String SQL = "delete from User where id = ?";
  jdbcTemplateObject.update(SQL, id);
  System.out.println("Borrado ID = " + id );
  return;
}

public void update(Integer id, Integer age){
  String SQL = "update User set age = ? where id = ?";
  jdbcTemplateObject.update(SQL, age, id);
  System.out.println("Actualizado ID = " + id );
  return;
}
```

7. Requisitos de uso

```java
import com.exampe.UserJDBCTemplate;
//
[...]

UserJDBCTemplate userTemplate = new UserJDBCTemplate();
userTemplate.setDataSource(getConnection());
```

Operaciones

```
userTemplate.create("Alberto", 30);

List<User> users = userTemplate.listUsers();
for (User u : users) {
  System.out.print("ID : " + u.getId());
  System.out.print(", Nombre : " + u.getName());
  System.out.println(", Edad : " + u.getAge());
}

userTemplate.update(1, 31);

User user = userTemplate.getStudent(1);
System.out.print("ID : " + user.getId() );
System.out.print(", Name : " + user.getName());
System.out.println(", Age : " + user.getAge());
```

⚠️ Tarea 14 ⚠️

+2 segundo parcial

- Integrar modelo de usuario en la aplicación web.

Fuentes

- https://www.tutorialspoint.com/spring/spring_jdbc_example.htm