

Adding a New Product in the Database

Documentation

Introduction

This documentation provides an overview and explanation of a simple Java web application for managing product details. The application uses Servlets, JavaServer Pages (JSP), and Hibernate for database interaction. The purpose of the application is to allow users to add product information, which is then stored in a MySQL database.

Table of Contents

- Project Structure
- Dependencies
- Database Configuration
- Servlets
- JSP Pages
- Hibernate Configuration
- Model Class
- Usage

1. Project Structure

The project follows a standard Java web application structure with the following main components:

- **src/com/example/servlet:** Contains Servlets, such as **AddProductServlet.java**.
- **WebContent:** Contains JSP pages (**addProduct.jsp**, **success.jsp**).
- **src/com/example/model:** Contains the **Product** class representing the data model.
- **hibernate.cfg.xml:** Hibernate configuration file.
- **WebContent/WEB-INF/lib:** Contains Hibernate and other necessary JAR files.
- **WebContent/WEB-INF/web.xml:** Deployment descriptor for configuring Servlets.

2. Dependencies

- **Java EE:** The project uses Java EE technologies, including Servlets and JSP.
- **Hibernate:** Object-relational mapping framework for database interaction.
- **MySQL Connector/J:** JDBC driver for MySQL database connectivity.

3. Database Configuration

The application connects to a MySQL database. Database configuration details, such as URL, username, and password, are specified in the **hibernate.cfg.xml** file.

4. Servlets

- **AddProductServlet:** Handles the HTTP POST request from the product addition form (**addProduct.jsp**). Retrieves form data, creates a **Product** object, and saves it to the database using Hibernate.

5. JSP Pages

- **addProduct.jsp:** HTML form for adding a new product. Sends data to the **AddProductServlet** for processing.
- **success.jsp:** Simple confirmation page displayed after successfully adding a product.

6. Hibernate Configuration

- **hibernate.cfg.xml:** Configures Hibernate settings, including database connection details, connection pool settings, and dialect. Also specifies the annotated entity class (**Product**).

7. Model Class

- **Product.java:** Represents the data model for a product. Annotated with JPA annotations (**@Entity**, **@Id**).

8. Usage

- Ensure that a MySQL database is set up with the specified configuration.
- Deploy the application to a Servlet container (e.g., Apache Tomcat).
- Access the application through the web browser.
- Fill out the product addition form on **addProduct.jsp** and submit.
- Upon successful addition, the user is redirected to **success.jsp**.