

JDBC

1. Cách set up dependency cho Dynamic Web Project
2. Tạo cơ sở dữ liệu cho Book
3. JDBC Servlet
4. Maven

Vào đường link <https://downloads.mysql.com/archives/c-j/> để download MySQL Java connector như bên dưới.

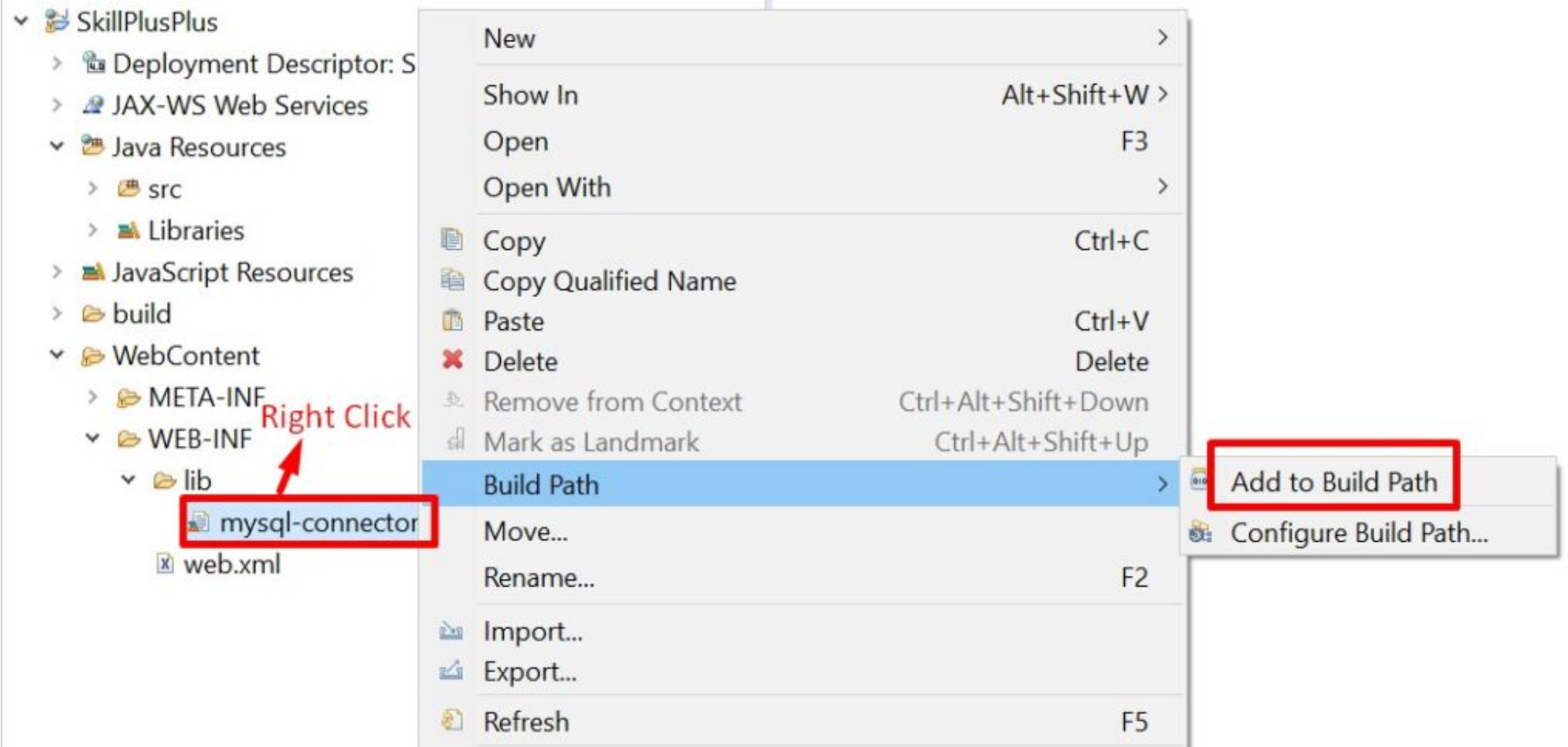
Connector/J 8.0.28

Select Operating System:

Platform Independent

Platform Independent (Architecture Independent), Compressed TAR Archive (mysql-connector-java-8.0.28.tar.gz)	8.0.28	4.0M	Download
Platform Independent (Architecture Independent), ZIP Archive (mysql-connector-java-8.0.28.zip)	8.0.28	4.8M	Download
			MD5: d19db89de75c46b71c4257badfa1e22b Signature

- ▼ SkillPlusPlus
 - > Deployment Descriptor: SkillPlusPlus
 - > JAX-WS Web Services
 - ▼ Java Resources
 - ▼ src
 - > skillplusplus
 - > Libraries
 - > JavaScript Resources
 - > Referenced Libraries
 - > build
 - ▼ WebContent
 - > META-INF
 - ▼ WEB-INF
 - ▼ lib
 - mysql-connector-java-8.0.28.jar
 - web.xml



Xây dựng ứng dụng nhập sách trong thư viện với các yêu cầu:

1. Tạo schema có tên là “jdbc_demo”
2. Tạo CSDL gồm 1 bảng có cấu trúc như sau:

Book

<u>STT</u>	<u>Tên trường</u>	<u>Kiểu dữ liệu</u>
1	<u>bookcode</u>	INT
2	<u>title</u>	TEXT
3	<u>author</u>	TEXT
4	<u>category</u>	TEXT
5	<u>approved</u>	INT

3. Thêm (INSERT) vào CSDL 2 dòng dữ liệu ví dụ tự chọn.
4. Đọc hết dữ liệu trong bảng Book (SELECT).
5. Cập nhật (UPDATE) 1 trong 2 dòng dữ liệu trên giá trị “category” và “approved”.
6. Xóa (DELETE) 1 trong 2 dòng dữ liệu trên

Những bước cơ bản để lấy dữ liệu bằng JDBC

- Import JDBC packages.
- Load and register the JDBC driver.
- Open a connection to the database.
- Create a statement object to perform a query.
- Execute the statement object and return a query resultset.
- Process the resultset.
- Close the resultset and statement objects.
- Close the connection.

These steps are described in detail in the sections that follow.

Tạo class Servlet MySQLDemo để truy cập CSDL Book đã tạo bên trên

```
package demoServlet;
```

```
import javax.servlet.annotation.WebServlet;
```

```
import javax.servlet.http.HttpServlet;
```

```
@WebServlet("/books")
```

```
public class MySQLDemo extends HttpServlet{
```

```
    //doget ở đây
```

```
    //dopost ở đây
```

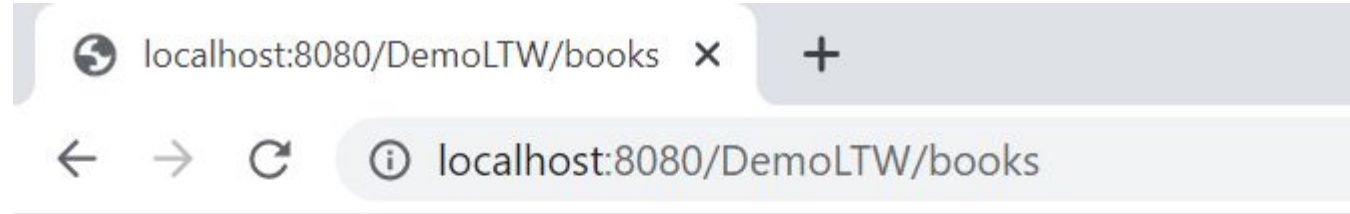
```
}
```

Bên dưới là hàm doGet để lấy tất cả những giá trị sách

@Override

```
protected void doGet(HttpServletRequest req,  
HttpServletRequest resp) throws ServletException, IOException {  
    resp.setContentType("text/html");  
    PrintWriter printer = resp.getWriter();  
    Connection connection=null;  
    Statement statement=null;  
    ResultSet resultSet=null;  
    try {  
        Class.forName("com.mysql.cj.jdbc.Driver");  
        connection =  
DriverManager.getConnection("jdbc:mysql://localhost:3306/jdbc_de  
mo",  
                                "root", "password");  
        statement = connection.createStatement();  
        resultSet = statement.executeQuery("select * from book");
```


Hàm doGet trả về giá trị như bên dưới



1, harry potter, j k rowling, null, 0
2, IT, Stephen King, horror, 1
3, Bible, Jesus?, sci-fi, 1

P/S: Các em có thể tìm hiểu về cách show giá trị trên dưới dạng bảng bằng JSP

Tạo 1 quyển sách mới vào trong CSDL = API sử dụng phương thức POST

Tạo 1 file AddBook.jsp như bên dưới

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="books" method="post">
BookCode: <input type="number" name="bookcode"></br>
Title: <input type="text" name="title"></br>
Author: <input type="text" name="author"></br>
Category: <input type="text" name="category"></br>
Approved: <input type="checkbox" name="approved"></br>
<input type="submit" value="Add Book">
</form>
</body>
</html>
```

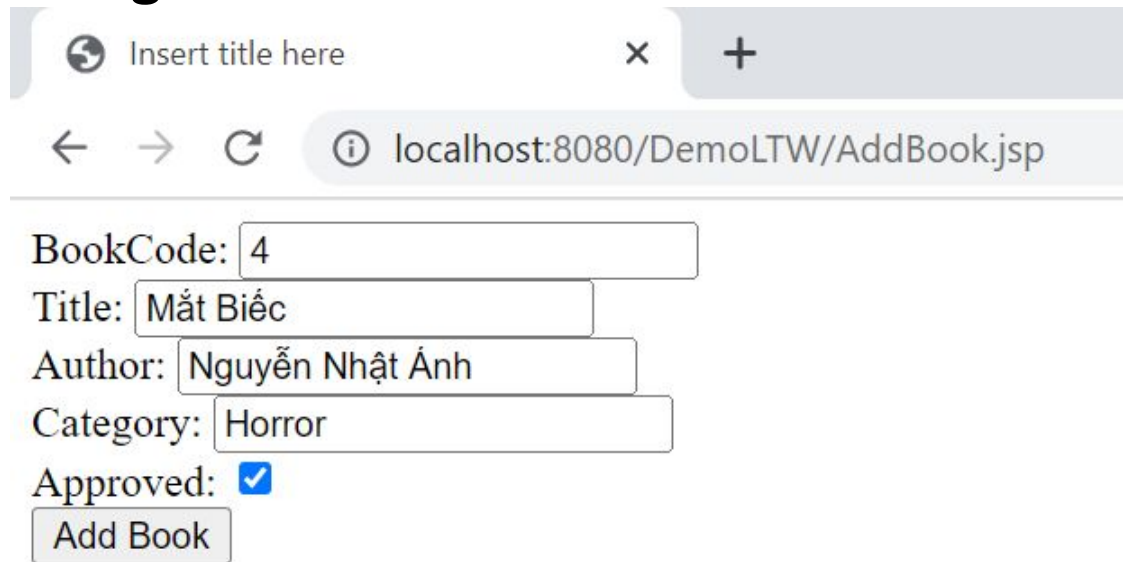
Bên dưới là hàm doPost để tạo mới 1 quyển sách

```
@Override
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
    resp.setContentType("text/html");
    Connection connection=null;

    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        connection =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/jdbc_demo", "root",
                "password");
        // Query/statement to insert the values
        PreparedStatement st = connection.prepareStatement("INSERT INTO book
VALUES (?, ?, ?, ?, ?)");

        // set values into the query
        st.setInt(1, Integer.valueOf(req.getParameter("bookcode")));
        st.setString(2, req.getParameter("title"));
        st.setString(3, req.getParameter("author"));
        st.setString(4, req.getParameter("category"));
        st.setInt(5, Integer.parseInt(req.getParameter("id")));
        st.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Trang thêm sách



Insert title here × +

← → ↻ ⓘ localhost:8080/DemoLTW/AddBook.jsp

BookCode: 4

Title: Mắt Biếc

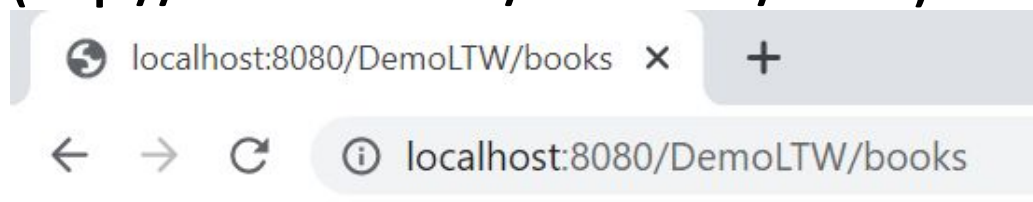
Author: Nguyễn Nhật Ánh

Category: Horror

Approved: ☒

Add Book

Sau khi thêm sách thành công website sẽ chuyển về trang danh sách sách ở bên trên (<http://localhost:8080/DemoLTW/books>)



localhost:8080/DemoLTW/books × +

← → ↻ ⓘ localhost:8080/DemoLTW/books

- 1, harry potter, j k rowling, null, 0
- 2, IT, Stephen King, horror, 1
- 3, Bible, Jesus?, sci-fi, 1
- 4, Mắt Biếc, Nguyễn Nhật Ánh, Horror, 1

Hàm doPost ở trên sử dụng Prepared Statement thay vì Statement bình thường như ở hàm doGet để dễ đưa giá trị parameter vào câu lệnh SQL hơn

Statement vs PreparedStatement



- Are these the same? What do they do?

```
String val = "abc";  
Statement stmt = con.createStatement( );  
ResultSet rs =  
    stmt.executeQuery("select * from R where A=" + val);
```

```
String val = "abc";  
PreparedStatement pstmt =  
    con.prepareStatement("select * from R where A=?");  
pstmt.setString(1, val);  
ResultSet rs = pstmt.executeQuery();
```

Hàm doPost ở trên chưa hề có bất kì 1 sự kiểm tra (validation) nào.

Luôn nhớ phải thực hiện validate dữ liệu nhận được:

- Báo lỗi khi có ô nhập dữ liệu bị để trống.
- Báo lỗi khi dữ liệu nhập vào bị sai định dạng.
- Không cho phép nhập trùng dữ liệu sách đã có trong CSDL.
- Nếu có lỗi quay trở về form nhập ban đầu và hiển thị lỗi cạnh ô nhập.
-

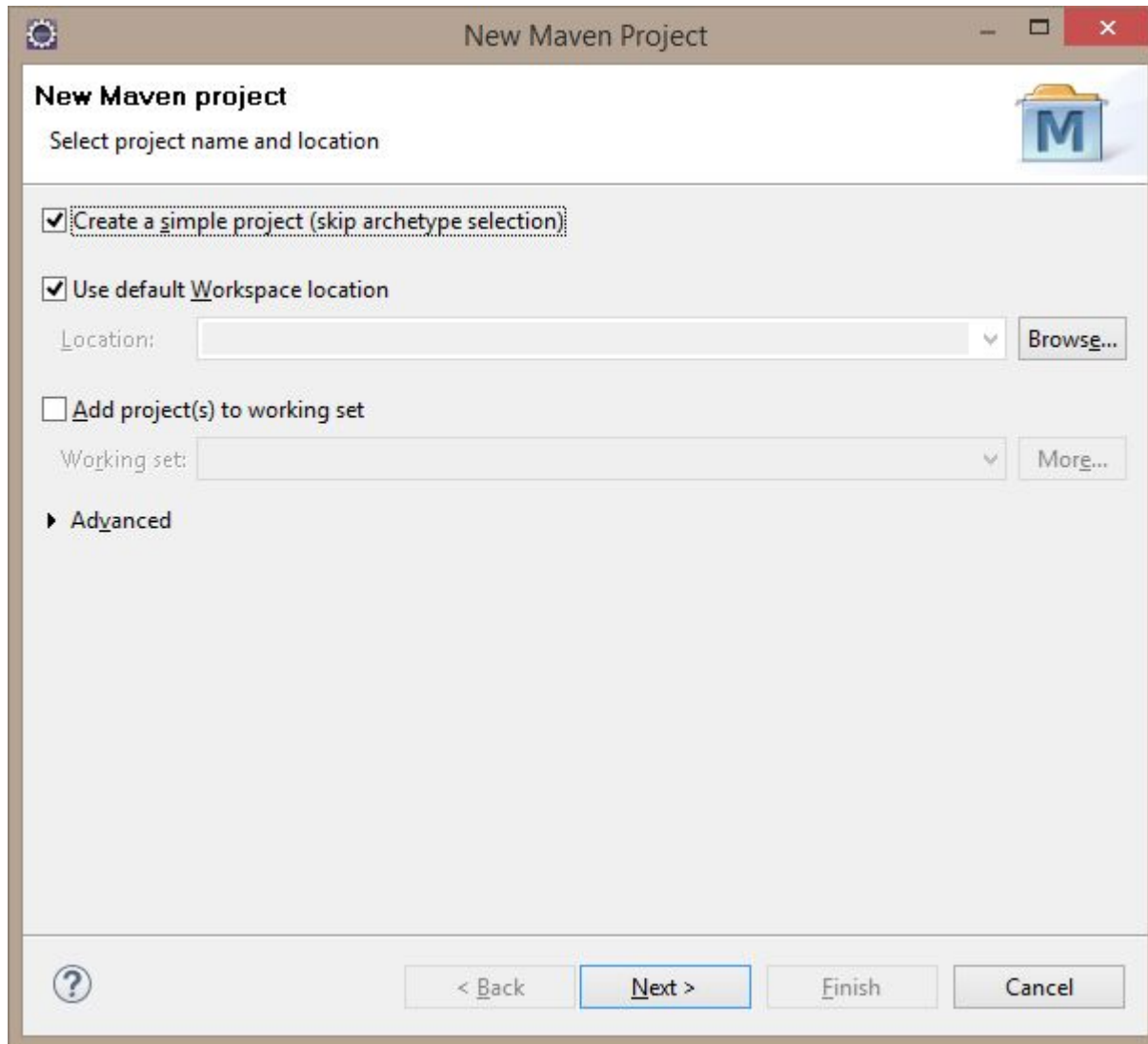
Maven

Maven là gì ?

Apache maven là một chương trình quản lý dự án cho phép các developers có thể quản lý về version, các dependencies (các thư viện sử dụng trong dự án) , **quản lý build**, tự động download javadoc & source, ...

Có thể tạo Project Maven mới hoặc covert Dynamic Web Project cũ sang Maven Project (choột phải vào tên project -> Configure -> Convert to Maven Project

Bước 1: File -> New -> Maven Project. Trong dialog hiện ra, lựa chọn “Create a simple project ...” như trong hình. Chọn next



New Maven Project

New Maven project

Configure project

Artifact

Group Id: com.example

Artifact Id: maven-sample

Version: 0.0.1-SNAPSHOT

Packaging: war

Name:

Description:

Parent Project

Group Id:

Artifact Id:

Version: Browse... Clear

Advanced

? < Back Next > Finish Cancel

Trong đó :

- Group Id : Tên tổ chức / công ty / cá nhân của dự án
- Artifact Id : Tên của package, dự án
- Version : version của project
- Package : để ý 2 giá trị : jar có nghĩa là thư viện or java application, war là web application
- Name : Tên project (trong eclipse)

Nhấn finish

Cấu hình file pom.xml

```
1<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
2    <modelVersion>4.0.0</modelVersion>
3    <groupId>com.ltw3</groupId>
4    <artifactId>DemoMaven</artifactId>
5    <version>0.0.1-SNAPSHOT</version>
6    <packaging>war</packaging>
7
8    <dependencies>
9        <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
10        <dependency>
11            <groupId>mysql</groupId>
12            <artifactId>mysql-connector-java</artifactId>
13            <version>8.0.28</version>
14        </dependency>
15        <!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
16        <dependency>
17            <groupId>javax.servlet</groupId>
18            <artifactId>javax.servlet-api</artifactId>
19            <version>4.0.1</version>
20            <scope>provided</scope>
21        </dependency>
22    </dependencies>
23 </project>
```

Thêm thẻ dependencies trước khi thêm các dependency

Copy từ trang Maven Repository

Sau khi bấm
Save file
Eclipse sẽ tự
động
download các
dependencies