



XÂY DỰNG ỨNG DỤNG CRUD VỚI JPA LẬP TRÌNH JAVA #4 (P2.1)

- Giới thiệu ứng dụng CRUD
- Tổ chức dự án CRUD
- Thiết kế giao diện (View) CRUD
- Thiết kế Servlet (Controller) CRUD



GIỚI THIỆU ỨNG DỤNG CRUD

Id:

Password:

Fullname:

Email Address:

Role:

Admin User

Create Update Delete Reset

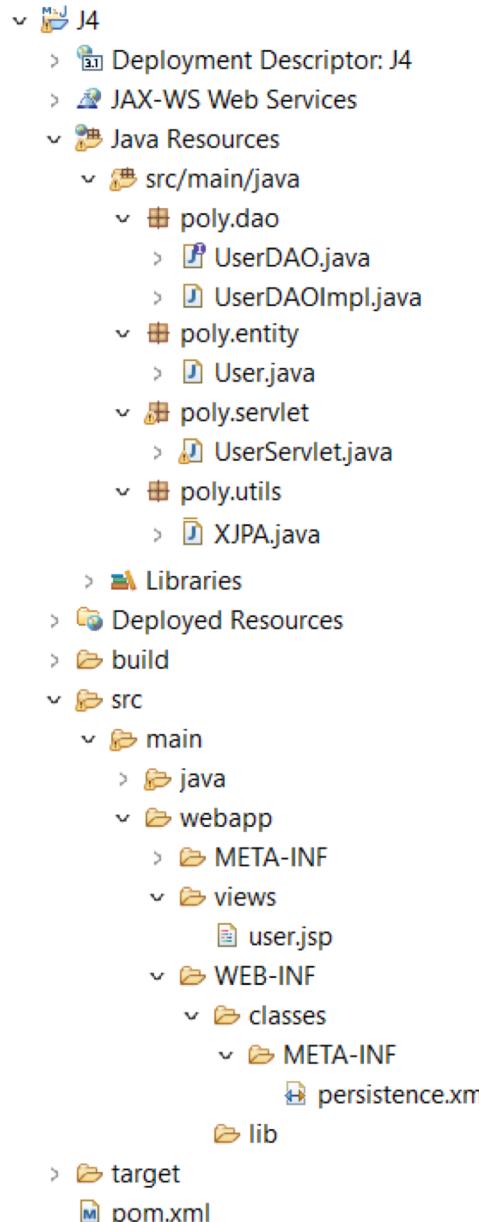
Id	Fullname	Email	Role	
			Admin/User	Edit

THANHVINHCENTER.PolyOE - dbo.Users

Column Name	Data Type	Allow Nulls
Id	nvarchar(20)	<input type="checkbox"/>
Password	nvarchar(50)	<input type="checkbox"/>
Fullname	nvarchar(50)	<input type="checkbox"/>
Email	nvarchar(50)	<input type="checkbox"/>
Admin	bit	<input type="checkbox"/>

GIỚI THIỆU ỨNG DỤNG CRUD

Sự kiện	Điều khiển sự kiện
Page.Open	<ul style="list-style-type: none"> Hiển thị bảng Xóa trắng form
Create.Click	<ul style="list-style-type: none"> Thêm mới Hiển thị lại bảng Xóa trắng form
Update.Click	<ul style="list-style-type: none"> Cập nhật Hiển thị lại bảng Giữ nguyên dữ liệu trên form
Delete.Click	<ul style="list-style-type: none"> Xóa mục theo id trên form Hiển thị lại bảng Xóa trắng form
Reset.Click	<ul style="list-style-type: none"> Xóa trắng form Hiển thị lại bảng
Edit.Click	<ul style="list-style-type: none"> Hiển thị mục chọn lên form Hiển thị lại bảng



TỔ CHỨC DỰ ÁN

□ MVC

- ❖ M (**poly.entity.User**): lớp thực thể mô tả cấu trúc dữ liệu của table *Users* và form.
- ❖ V (**views/user.jsp**): giao diện
- ❖ C (**poly.servlet.UserServlet**): điều khiển các yêu cầu được gửi từ trình duyệt

□ DAO

- ❖ **poly.dao.UserDAO** là interface khai báo các phương thức làm việc cụ thể với entity *User*
- ❖ **poly.dao.UserDAOImpl** là class thực hiện theo *UserDAO*
- ❖ **poly.utils.XJPA**: Lớp tiện ích cung cấp các phương thức EntityManager() dùng chung trong các DAOImpl.
- ❖ **WEB-INF/classes/META-INF/persistence.xml**: Khai báo thông số kết nối đến CSDL

□ Dependences (**pom.xml**)

- ❖ SQL Server Driver, JSTL, Hibernate/JPA, BeanUtils

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1"
    xmlns="http://xmlns.jcp.org/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
    <persistence-unit name="PolyOE">
        <properties>
            <property name="javax.persistence.jdbc.driver"
                value="com.microsoft.sqlserver.jdbc.SQLServerDriver" />
            <property name="javax.persistence.jdbc.url"
                value="jdbc:sqlserver://localhost;database=PolyOE" />
            <property name="javax.persistence.jdbc.user" value="sa" />
            <property name="javax.persistence.jdbc.password" value="*****" />
        </properties>
    </persistence-unit>
</persistence>
```

*Hiệu chỉnh các thông số kết nối phù hợp với môi trường của bạn
Giảng viên lấy code ở phần notes*

```
<!--Hibernate/JPA-->
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>6.4.2.Final</version>
</dependency>
<!--SQL Server Driver-->
<dependency>
    <groupId>com.microsoft.sqlserver</groupId>
    <artifactId>mssql-jdbc</artifactId>
    <version>12.8.0.jre8</version>
</dependency>
```

```
<!--BeanUtils-->
<dependency>
    <groupId>commons-beanutils</groupId>
    <artifactId>commons-beanutils</artifactId>
    <version>1.9.3</version>
</dependency>
<!--JSTL-->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
</dependency>
```

- User là lớp mô tả cấu trúc dữ liệu của bảng Users được sử dụng để buộc với form và các hàng trên table trên giao diện

```
@Entity  
@Table(name = "Users")  
public class User {  
    @Id  
    private String id;  
    private String password;  
    private String email;  
    private String fullname;  
    private Boolean admin = false;  
    getters/setters  
}
```



Column Name	Data Type	Allow Nulls
Id	nvarchar(20)	<input type="checkbox"/>
Password	nvarchar(50)	<input type="checkbox"/>
Fullname	nvarchar(50)	<input type="checkbox"/>
Email	nvarchar(50)	<input type="checkbox"/>
Admin	bit	<input type="checkbox"/>

```
<%@ page pageEncoding="utf-8"%>
<%@ taglib uri="http://java.sun.com/jstl/core_rt" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>User Management</title>
</head>
<body>
    <c:url var="path" value="/user" scope="request" />
    <!-- FORM -->
    <hr>
    <!-- TABLE -->
</body>
</html>
```

```
<form method="post">
    <input name="id" value="${item.id}"><br>
    <input name="password" value="${item.password}"><br>
    <input name="fullname" value="${item.fullname}"><br>
    <input name="email" value="${item.email}"><br>
    <input name="admin" type="radio" value="true" ${item.admin?'checked':''}> Admin
    <input name="admin" type="radio" value="false" ${item.admin?:"checked"}> User<br>
    <hr>
    <button formaction="${path}/create">Create</button>
    <button formaction="${path}/update">Update</button>
    <button formaction="${path}/delete">Delete</button>
    <button formaction="${path}/reset">Reset</button>
</form>
```

* \${item} là đối tượng **User** được chia sẻ từ **UserServlet**

XÂY DỰNG GIAO DIỆN USER.JSP - <TABLE>

```
<table border="1" style="width: 100%">
    <thead>...</thead>
    <tbody>
        <c:forEach var="d" items="${list}" varStatus="vs">
            <tr>
                <td>${vs.count}</td>
                <td>${d.id}</td>
                <td>${d.fullname}</td>
                <td>${d.email}</td>
                <td>${d.admin?'Admin':'User'}</td>
                <td><a href="#">path/edit/${d.id}>Edit</a></td>
            </tr>
        </c:forEach>
    </tbody>
</table>
```

* \${list} là **List<User>** được chia sẻ từ **UserServlet**

TỔNG KẾT CÁC TƯƠNG TÁC TỪ NGƯỜI DUNG LÊN GIAO DIỆN

Event	Request
Page.Open	GET:\${path}/index
Create.Click	POST:\${path}/create
Update.Click	POST:\${path}/update
Delete.Click	POST:\${path}/delete
Reset.Click	POST:\${path}/reset
Edit.Click	GET:\${path}/edit/{id}

<c:url var="**path**" value="/user" scope="request" />

XÂY DỰNG USERSERVLET – TỔNG THỂ

```
@WebServlet({  
    "/user/index",  
    "/user/edit/*",  
    "/user/create",  
    "/user/update",  
    "/user/delete",  
    "/user/reset"  
})  
public class UserServlet extends HttpServlet {  
    @Override  
    protected void service(HttpServletRequest req, HttpServletResponse resp)  
        throws ServletException, IOException {  
        User form = new User();  
        //--code xử lý--//  
        req.setAttribute("item", form);  
        List<User> list = dao.findAll();  
        req.setAttribute("list", list);  
        req.getRequestDispatcher("/views/user.jsp").forward(req, resp);  
    }  
}
```

```
try {  
    BeanUtils.populate(form, req.getParameterMap());  
} catch (IllegalAccessException | InvocationTargetException e) {  
    e.printStackTrace();  
}  
UserDAO dao = new UserDAOImpl();  
String path = req.getServletPath();  
if (path.contains("edit")) {  
    String id = req.getPathInfo().substring(1);  
} else if (path.contains("create")) {  
} else if (path.contains("update")) {  
} else if (path.contains("delete")) {  
} else {  
    form = new User();  
}
```

Đọc dữ liệu form

Phân biệt request theo URL



DEMOSTATION

- ✓ Giới thiệu ứng dụng CRUD
- ✓ Tổ chức dự án CRUD
- ✓ Thiết kế giao diện (View) CRUD
- ✓ Thiết kế Servlet (Controller) CRUD





Cảm ơn