



FPT POLYTECHNIC



Conceive Design Implement Operate

ỨNG DỤNG MÔ HÌNH CRUD VÀ JDBC

LẬP TRÌNH JAVA #3 (P7.1)

www.poly.edu.vn

- ❑ Giới thiệu ứng dụng CRUD
- ❑ Tổ chức ứng dụng
- ❑ Xây dựng giao diện JSP
- ❑ Xây dựng Servlet tiếp nhận các yêu cầu từ giao diện



GIỚI THIỆU ỨNG DỤNG CRUD

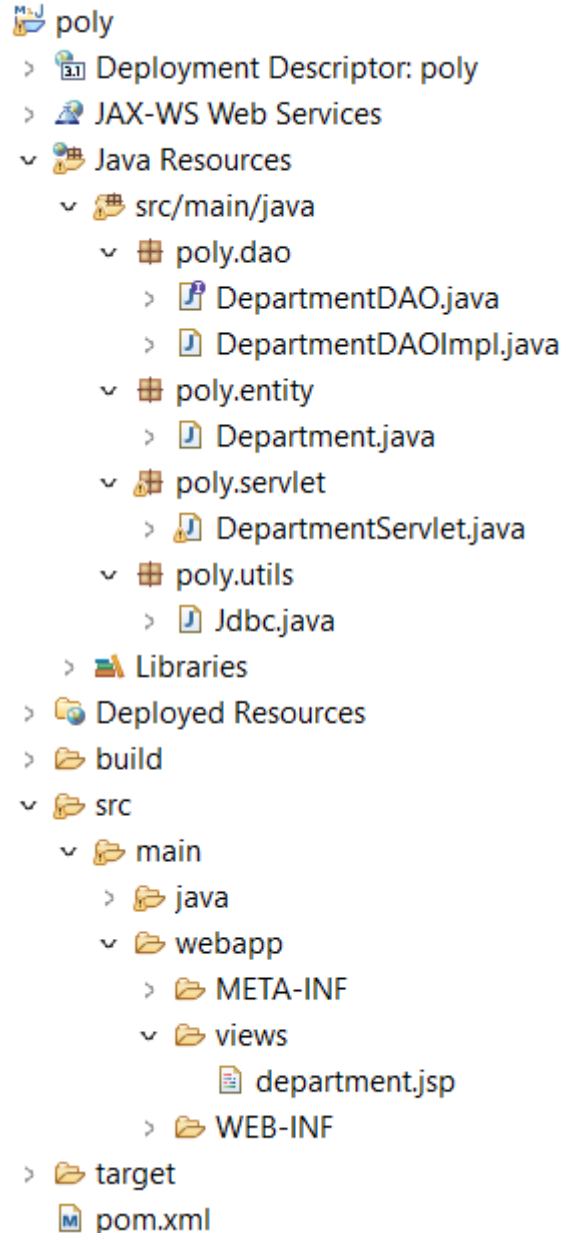
Id:

Name:

Description:

No.	Id	Name	Description	
				Edit

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"> Hiển thị lại bảng Xóa trắng form
Edit.Click	<ul style="list-style-type: none"> Hiển thị mục chọn lên form Hiển thị lại bảng



❑ MVC

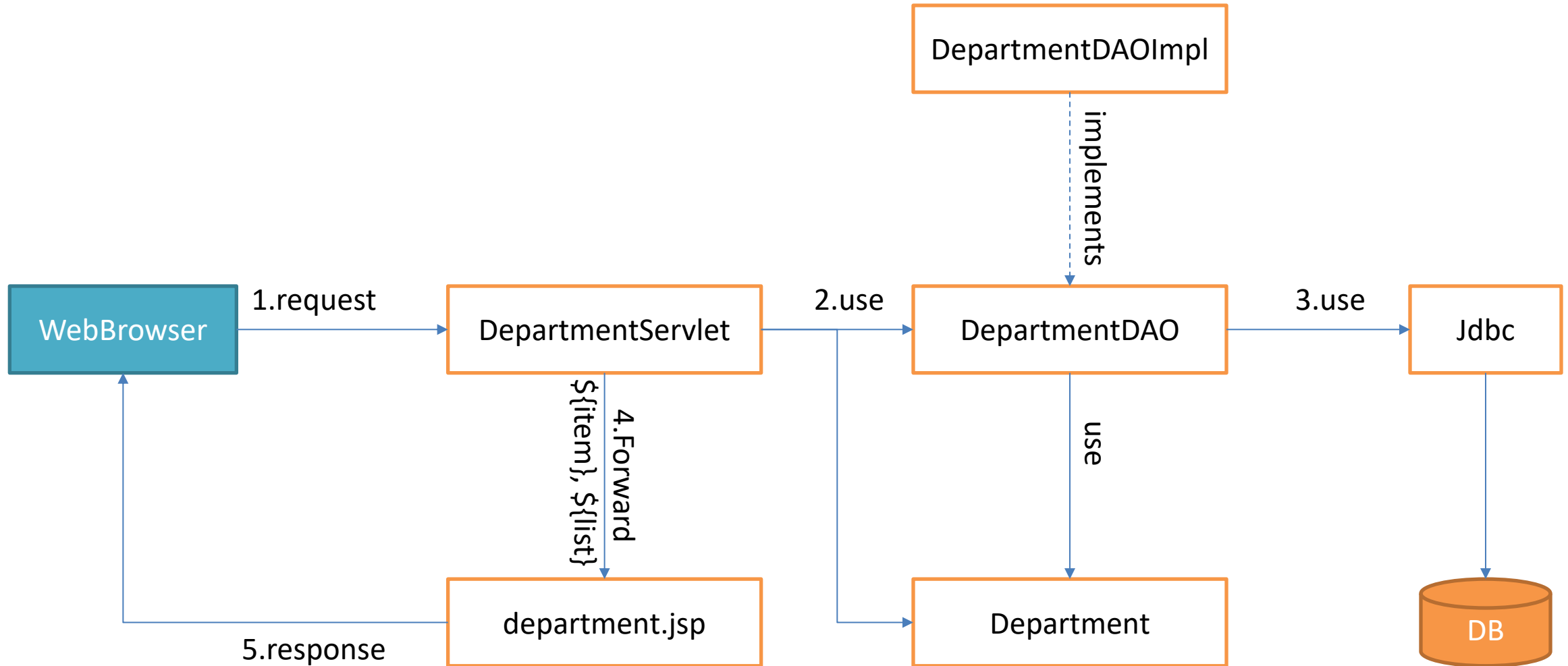
- ❖ M (Department): lớp thực thể mô tả cấu trúc dữ liệu của table Departments và form.
- ❖ V (Department.jsp): giao diện
- ❖ C (DepartmentServlet): điều khiển các yêu cầu được gửi từ trình duyệt

❑ DAO

- ❖ Jdbc: Lớp tiện ích cung cấp các phương thức dùng chung thao tác và truy vấn dữ liệu từ CSDL.
- ❖ DepartmentDAO (Thiết kế) là interface khai báo các phương thức làm việc cụ thể với bảng Departments
- ❖ DepartmentDAOImpl (Thi công) là class thực hiện theo DepartmentDAO

❑ Dependences (pom.xml)

- ❖ SQL Server Driver
- ❖ JSTL



- ❑ Department là lớp mô tả cấu trúc dữ liệu của một phòng ban được sử dụng
 - ❖ Buộc với form và các hàng trên table trên giao diện
 - ❖ Làm việc với bảng Departments trong CSDL
- ❑ Department phải thỏa mãn quy ước của một Bean Class

```
public class Department {  
    private String id;  
    private String name;  
    private String description;  
    //--getters/setters--//  
}
```



Column	Type
Id	VARCHAR(20)
Name	NVARCHAR(50)
Description	NVARCHAR(50)

```
<%@ page pageEncoding="utf-8"%>
<%@ taglib uri="http://java.sun.com/jstl/core_rt" prefix="c" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Department Management</title>
</head>
<body>
    <c:set var="path" value="{pageContext.request.contextPath}/department"/>
    <!-- FORM -->
    <form method="post">...</form>
    <hr>
    <!-- TABLE -->
    <table border="1" style="width:100%">...</table>
</body>
</html>
```

XÂY DỰNG GIAO DIỆN DEPARTMENT.JSP - <FORM>

```
<form method="post">
  <label>Id:</label><br>
  <input name="id" value="\${item.id}"><br>
  <label>Name:</label><br>
  <input name="name" value="\${item.name}"><br>
  <label>Description:</label><br>
  <textarea name="description" rows="3" cols="40">\${item.description}</textarea>
  <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 **Department** được chia sẻ từ *DepartmentServlet*

XÂY DỰNG GIAO DIỆN DEPARTMENT.JSP - <TABLE>

```
<table border="1" style="width:100%">
  <thead> <!-- tiêu đề table --> </thead>
  <tbody>
    <c:forEach var="d" items="${list}" varStatus="vs">
      <tr>
        <td>${vs.count}</td>
        <td>${d.id}</td>
        <td>${d.name}</td>
        <td>${d.description}</td>
        <td><a href="${path}/edit/${d.id}">Edit</a></td>
      </tr>
    </c:forEach>
  </tbody>
</table>
```

* *\${list}* là **List<Department>** được chia sẻ từ *DepartmentServlet*

TỔNG KẾT CÁC TƯƠNG TÁC TỪ NGƯỜI DÙNG LÊN GIAO DIỆN

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

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

XÂY DỰNG DEPARTMENTSERVLET – CODE XỬ LÝ

```
try {  
    BeanUtils.populate(form, req.getParameterMap());  
} catch (IllegalAccessException | InvocationTargetException e) {  
    e.printStackTrace();  
}  
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 Department();  
}
```

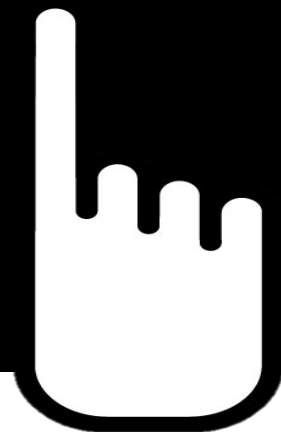
Đọc dữ liệu form

Phân biệt request theo URL



DEMO

THỰC HIỆN DỰ ÁN VÀ CHẠY THỬ



- ✓ Giới thiệu ứng dụng CRUD
- ✓ Tổ chức ứng dụng
- ✓ Xây dựng giao diện JSP
- ✓ Xây dựng Servlet tiếp nhận các yêu cầu từ giao diện





Cảm ơn