

สรุป INT303 Web Programming [LAB]

Preparing

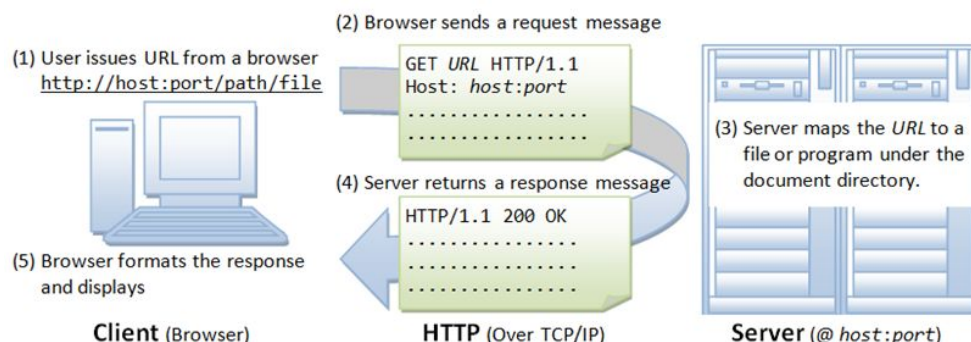
1. java : syntax, conditional, datatypes, OOP, how to java connect db [use controller]
2. Web : html [form, div, table], css [inline]
3. Database : create, insert, delete, update

เนื้อหา Web Programming

1. Overview
2. การสร้าง Web Application
3. Servlet
 - โครงสร้าง
 - การ GET, SET และส่งค่าไปยังที่อื่น
 - session
 - cookie
 - การดึงข้อมูลจาก DB ผ่าน JPA
4. JSP
 - โครงสร้าง
 - Form
 - JSP syntax elements, EL
5. Create and Connect DB with JPA
 - Persistence Unit
 - Entity Class
 - JPA Controller
6. Filter
7. การจัดการปัญหา ERROR

Overview

- กระบวนการ Request มี 2 แบบหลักๆ คือ GET [Query String], POST [อยู่ใน Protocol]
- กระบวนการ Response คือการตอบกลับจาก Server



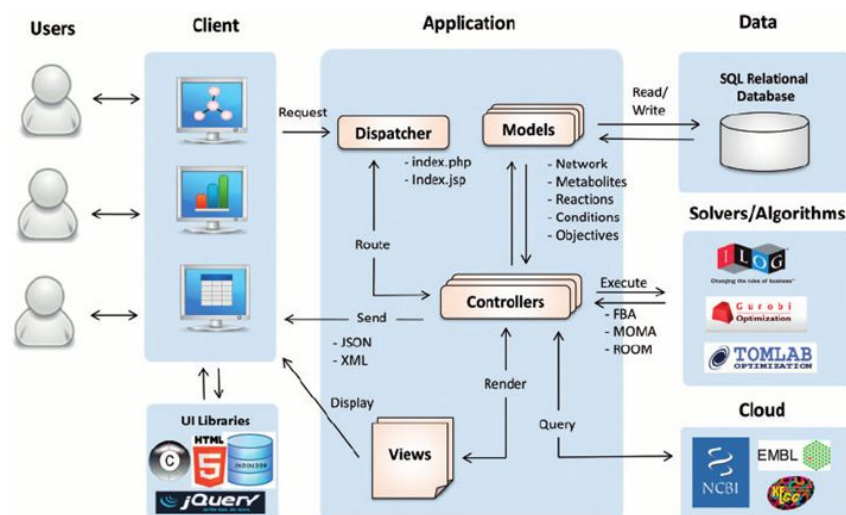
MVC Model

❖ องค์ประกอบ

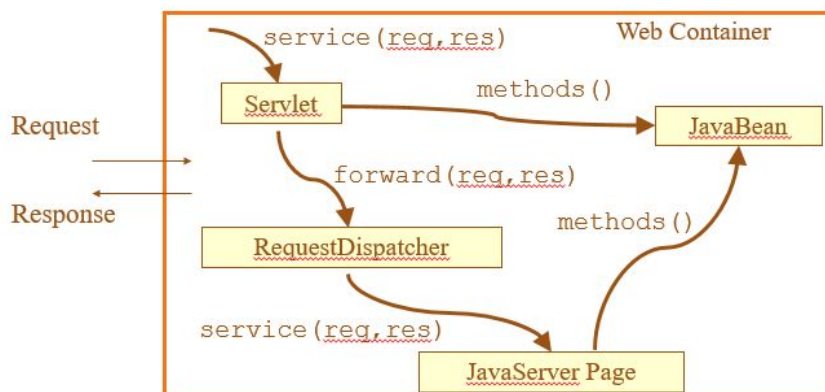
- 1.] Model: Java Class ต่างๆ
- 2.] View: JSP
- 3.] Controller: Servlet

❖ การทำงาน

- Request เข้ามาที่ Controller [ตรวจสอบ/เช็คข้อมูลว่าครบไหม เช่น Field ต่างๆ]
- Controller จะ...
 - ส่งไปที่ View เพื่อ Display
 - ส่งไปที่ Models [หัวใจสำคัญของ WebApp] เพื่อทำงานส่วนต่างๆ เช่น ดูเงื่อนไข, เขียนลง Database



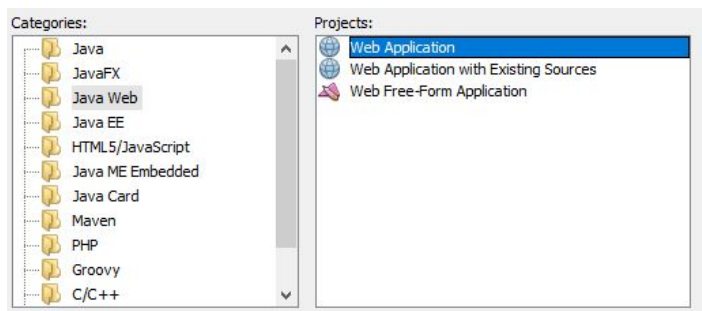
Web application evolution: JSP pages



PROJECT

Create Project

1. New Project / Java Web / Web Application

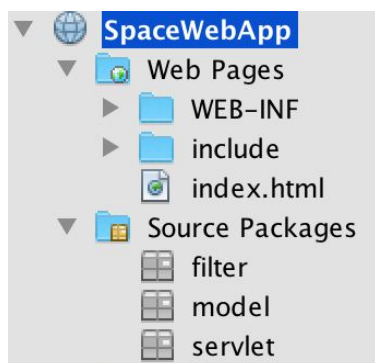


โดยตั้ง Context Path: /ชื่อที่เราอ้างถึงเวลาDeployServer

ตัวอย่าง localhost:8080/Space/

2. จะได้ Project ให้คลิกขวาเพื่อ Deploy และกด Run

3. โครงสร้าง Project



- Web Pages
 - WEB-INF : เก็บ web.xml
- Source Packages
 - filter
 - model
 - servlet

หน้า Index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>SIT SPACE</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- link bootstrap-->
  </head>
  <body>
    <style>
      body{
        font-family: sans-serif;
      }
    </style>
    <div>
      <h1>SIT SPACE</h1>
    </div>
    <div>
      <a href="ServletName">Name</a>
    </div>
  </body>
</html>
```

การสร้าง Servlet [Controller]

1. New File / Servlet ให้ตั้งชื่อ File ลงท้ายด้วย Servlet เช่น WeightConverterServlet
2. ให้ Add information ลง web.xml และตั้ง URL Pattern เป็น /WeightConverter
3. โครงสร้างไฟล์

```
public class WeightConverterServlet extends HttpServlet {

    /** Processes requests for both HTTP GET and POST ...9 lines */

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        //code
    }

    HttpServlet methods. Click on the + sign on the left to edit the code.
}
```

การ GET, SET และส่งค่าไปยังที่อื่น

- GET ค่า
ตัวแปร = request.getParameter("x") จะดึงค่ามาจาก <input name="x" > ใน view
**รับเข้ามาเป็น String หากเก็บเป็นตัวเลขควรใช้ Integer.parseInt() หรือ .valueOf();
- SET ค่า
request.setAttribute("ที่จะส่งชื่อไป", ตัวแปร);
- ส่งค่าไปยังที่อื่น
แบบ 1: forward
getServletContext().getRequestDispatcher("/ชื่อ.jsp").forward(request, response);
อธิบาย : RequestDispatcher คือตัวที่อนุญาตให้ forward request จาก servlet อื่นมาทำงานได้ โดย path จะเริ่มจาก root ซึ่งถ้าเรียกที่หน้าเดิมจะไม่เปลี่ยนเนื่องจากการทำงานฝั่ง Server
แบบ 2: sendredirect
response.sendRedirect("ชื่อServlet");
อธิบาย : จะ response กลับไปหา client ซึ่งถ้าเรียกที่หน้าเดิม url ก็จะเปลี่ยนเหมือนกับการคลิก link ใหม่ ดังนั้น Path จะนับจากตำแหน่งที่อยู่ปัจจุบัน
เช่น AddItemToCartServlet, RemoveItemFromCartServlet, ChooseBackgroundServlet

การดึงข้อมูลใส่ List

- สร้าง List
String[] subjects = request.getParameterValues('subjects');
List<String> subjectList = new ArrayList();
for (String subject : subjects) {
 subjectList.add(subject);
}

การใช้ session

- สร้าง Session

```
HttpSession session = request.getSession(true);           //true-ไม่มีสร้างใหม่
String textMessage = request.getParameter("message");
ObjectTypes objX = (ObjectTypes) session.getAttribute("x"); // down casting
if (objX==null) {                                           //ถ้ายังไม่มีค่า
    //-----ในกรณีต้องการสร้าง Session-----
    objX = new ObjectTypes ();                             //ทำให้เป็น Obj
    session.setAttribute("x", objX);                       //set session
}
objX.setMessage(textMessage)                               //method
- ให้ Session Invalidate
session.invalidate();
```

การใช้ Cookie

- สร้าง Cookie

```
Cookie ck = new Cookie("ชื่อ", ค่า);
ck.setMaxAge(60*min*hour*day);                             //หน่วยเป็นวินาที
response.addCookie(ck);
```

การดึงข้อมูลจาก Database

- เขียน Annotation

```
@PersistenceUnit (unitName = "dbPU")
EntityManagerFactory emf;
```

```
@Resource
```

```
UserTransaction utx;
```

- สร้าง Controller

```
TablenameJpaController ctrl = new TablenameJpaController(utx, emf);
```

- method ที่ใช้

1. เพิ่ม record เข้า DB:	ctrl.create(object)	[INSERT]
2. ลบ record ออกจาก DB:	ctrl.destroy(id);	[DELETE]
3. แก้ไขข้อมูลใน record:	ctrl.edit(object);	[UPDATE]
4. ค้นหา record ใน DB:	ctrl.findName(id);	[SELECT ONE]
5. เอาข้อมูลทั้งหมด	ctrl.findNameEntities();	[SELECT ALL]

** เช่น `List<Product> products = productJpaCtrl.findProductEntities();`

- หากต้องใช้ try-catch ให้อยู่ใน blocked scoped นั้น

การสร้าง JSP [View]

1. New File / JSP ซึ่งจะได้นามสกุลไฟล์เป็น .jsp
2. ตัวอย่างโครงสร้างไฟล์

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>SIT SPACE</title>
  </head>
  <body>
    <style>
      h1{
        text-align: center;
      }
    </style>
    <h1>Weight Converter</h1>
    <form action="ชื่อServlet" method="post">

    </form>
  </body>
</html>
```

การใช้ FORM

- action ห้ามใส่ “/” นำหน้า เพราะมันจะไปเรียกที่ WebServer แทน AppServer
- Tag ที่ใช้

<input type= “text”	name=“username” required>
<input type= “password”	name=“username” required>
<input type= “submit”	value=“ชื่อที่แสดงบนปุ่ม”>
<input type= “radio”	name=“gender” value=“male”>
<input type= “checkbox”	name=“subject” value=“GEN”>

การใช้ JSP EL

- รูปแบบการเขียน : `${ ชื่อ.เมธอด }` เช่น `${user.firstName}`
เช่น `<h2> Hello, ${user.lastName} </h2>`
- หาก EL พังลองตรวจสอบ `<%@ taglib prefix='c' uri='.....' %>` บสรหัส
- การเช็คเงื่อนไข
แบบ 1 :
`${เงื่อนไข ? “จริง” : “เท็จ”}` เช่น `${value == '#050E54' ? 'checked': ''}`
แบบ 2 :
`<c:if test=“${message !=null}”>`
`<p>Welcome, member!</p>`
`</c:if>`

แบบ 3 :

<c:choose>

<c:when test='\${user.role == 'member'}'>

<p>Welcome, memberName!</p>

</c:when>

<c:otherwise>

<p>Welcome, guest!</p>

</c:otherwise>

</c:choose>

- การใช้ Loop

<c:forEach items= "\${products}" var= "p" varStatus= "vs" >

<tr>

<td>\${vs.count}</td>

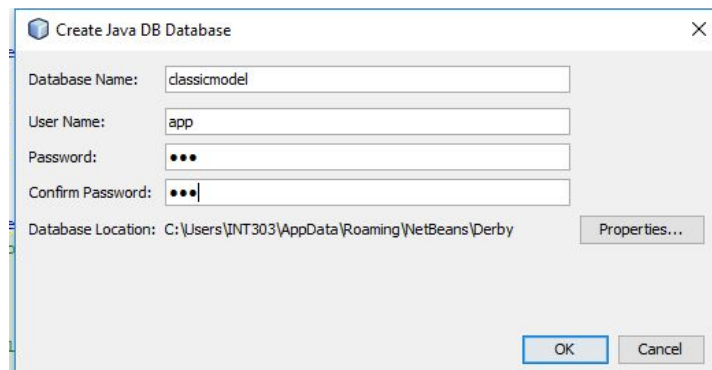
<td>\${p.name}</td>

</tr>

</ c:forEach >

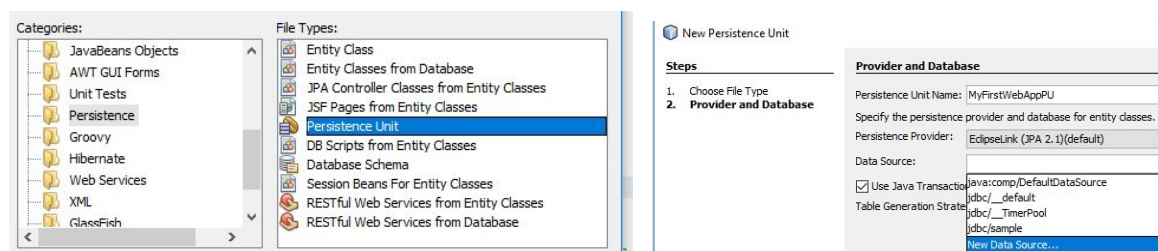
การสร้างและเชื่อมต่อ DB ผ่าน JPA

1. สร้าง Java DB Database



2. create table

3. สร้าง Persistence Unit / โดย Datasource ให้สร้างใหม่และเลือก DB ที่ต้องการ Connect

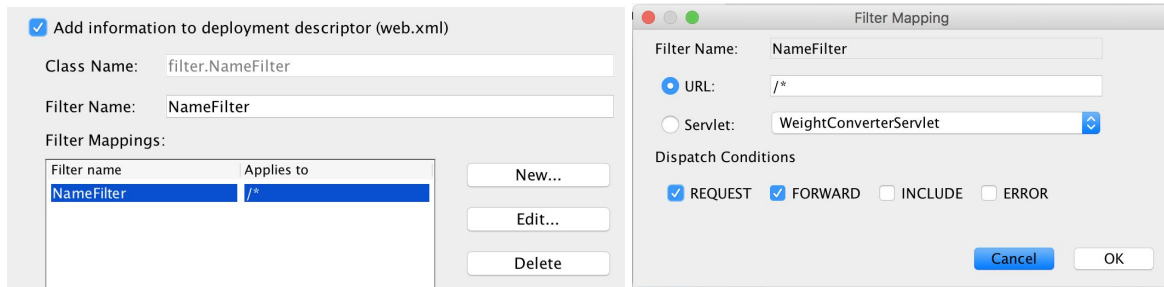


**เช่น JNDI Name: jdbc/classicmodel

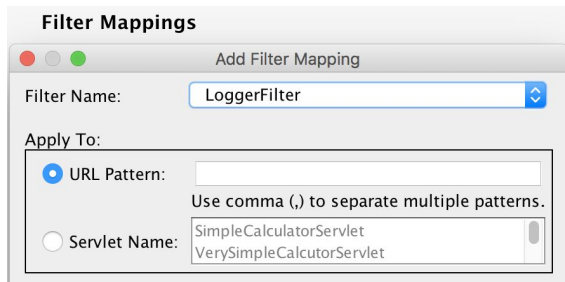
4. สร้าง Entity Class / Add ตารางที่ใช้ / Collection Type กำหนดเป็น List
5. สร้าง JPA Controller จาก Entity Classes

การทำ Filter

1. New File / Filter และตั้งชื่อ NameFilter
2. add to web.xml โดย mapping ว่าจะใช้กับServlet ตัวไหน [edit] **ถ้า /* คือทุก filter



3. เมื่อสร้างเสร็จสามารถ add ให้ไป mapping กับ Servlet ตัวไหนเพิ่มได้ใน web.xml



4. โครงสร้างไฟล์ (ให้ลบ method ที่หมดและ implements มา)

```
public class LoginFilter implements Filter {

    private FilterConfig config;

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {
        this.config = filterConfig;
    }

    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws ServletException, IOException {
        //do something before
        chain.doFilter(request, response);
        //do something when comeback
    }

    @Override
    public void destroy() {
    }

}
```

- การดึงข้อมูล ผ่าน [HttpServletRequest] request

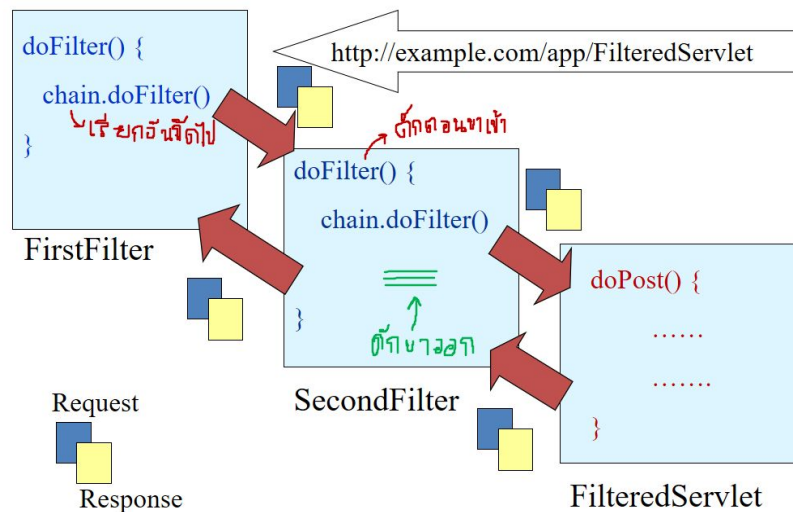
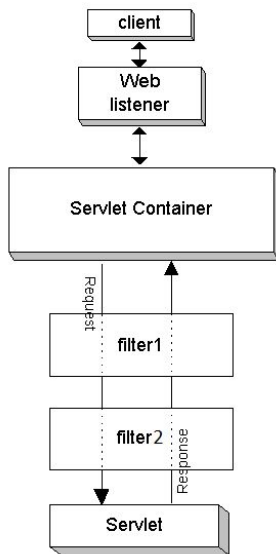
1. Session

HttpSession session = [(HttpServletRequest) request].getSession(false);

2. url

String url = [(HttpServletRequest)request].getRequestURI();

- การทำงานของ filter



ERROR

404 Page Not Found

- อาจเกิดจาก action ที่ใส่ชื่อผิด หากหน้าไม่เจอ

500 Internal Server Error

- อาจเกิดจากการที่ getParameter() แล้วไม่มี
แก้ไข: เช็คเงื่อนไขว่า ==null หรือไม่

ตัวอย่าง 1 : getParameter

Servlet

```

protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    String name = request.getParameter("name");
    String weighttext = request.getParameter("weight");
    double weight = Double.parseDouble(weighttext);

    request.setAttribute("name", name);
    request.setAttribute("weight", weight);

    getServletContext().getRequestDispatcher("/WeightConverterView.jsp").forward(request, response);
}
  
```

JSP

```

<h1>Weight Converter</h1>
<form action="WeightConverter" method="post">
  Name: <input type="text" name="name" /><br>
  Weight: <input type="number" name="weight" /> kg<br>
  <input type="submit" value="ENTER">
</form>
<div>
  <h5>DATA THAT WE GET:</h5>
  <p>
    NAME: ${name}<br>
    WEIGHT: ${weight} kg
  </p>
</div>
  
```

ตัวอย่าง 2 : Session

Servlet

```
HttpSession session = request.getSession(true);
request.setAttribute("name", name);
request.setAttribute("weight", weight);
Astronomer ast = (Astronomer) session.getAttribute("ast");
if(ast==null){
    ast = new Astronomer();
    session.setAttribute("ast", ast);
}
ast.setName(name);
ast.setWeight(weight);
```

JSP

```
<div>
  <h5>DATA THAT WE GET:</h5>
  <p>
    NAME: ${ast.name}<br>
    WEIGHT: ${ast.weight} kg<br>
    <b>Convert = ${ast.result}</b>
  </p>
</div>
```

ตัวอย่าง 3 : Authentication Filter

```
public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws
HttpSession session = ((HttpServletRequest)request).getSession(false); //If not have session
if(session==null||session.getAttribute("user")==null){ //if cart not have user
    config.getServletContext().getRequestDispatcher("/Login").forward(request, response);
    return;
}else{
    chain.doFilter(request, response);
}
}
```