

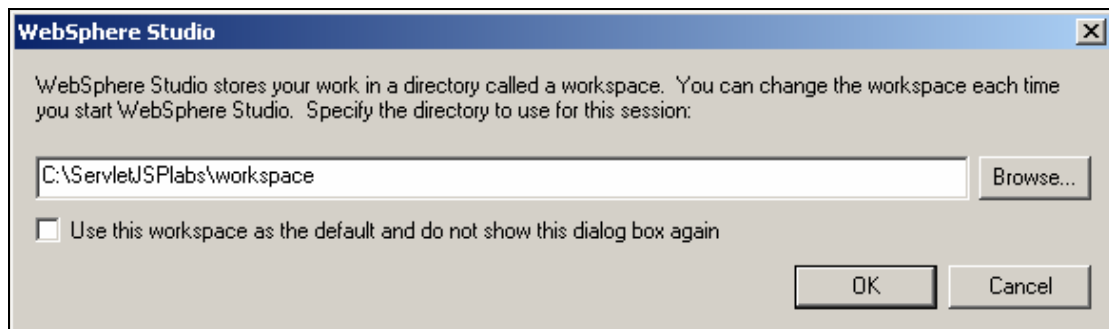
Exercise 1. Simple Registration Servlet

บทนำ

ในตัวอย่างนี้ เป็นการศึกษาในเรื่องการสร้าง Servlet ที่สืบทอดมาจาก คลาส `javax.servlet.http.HttpServlet` โดยทำการ override (สืบทอด) เมธอด `doGet` และ `doPost` โดย เมธอดทั้งสอง จะทำการเรียกใช้งาน เมธอดอื่น ที่ชื่อ `processRequest` ซึ่งมีกระบวนการสร้างเอกสาร HTML กลับไปยังหน้าจอผู้ใช้งาน เป็นข้อความยืนยัน ที่แสดงค่าของ `patron id`

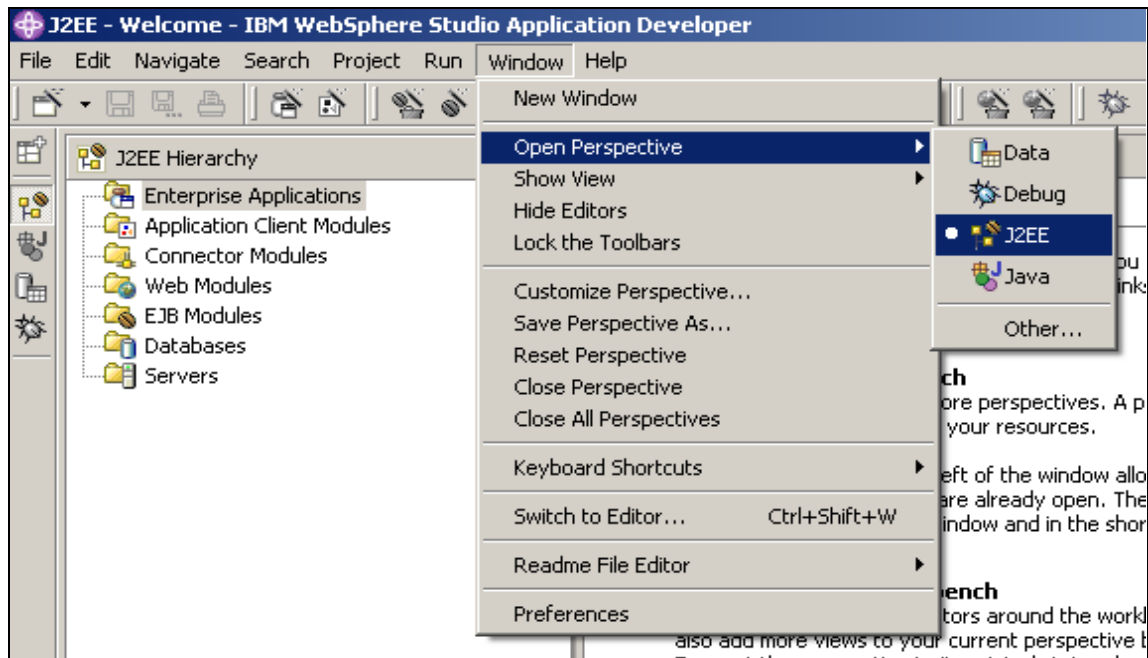
ขั้นตอนที่ 1 สร้าง Enterprise application และ Web projects

1. เปิดโปรแกรม WebSphere Studio Application Developer ผ่านกระบวนการดังนี้
 - เลือกที่ เมนู **Start->Programs->IBM WebSphere Studio->Application Developer 5.1.1**
 - กำหนดค่า “**C:\ServletJSPlabs\workspace**” ให้เป็นพื้นที่ที่ใช้ในการพัฒนา (workspace) ดังรูปที่ 1 หรือ อาจทำการคลิกปุ่ม **Browse...** เพื่อเลือก workspace เมื่อเสร็จแล้ว ให้ทำการคลิกปุ่ม **OK** เพื่อยืนยัน



รูปที่ 1 แสดงหน้าต่างกำหนดค่า workspace

2. ในการเปิดใช้งาน WebSphere Studio Application Developer ในครั้งแรก ที่หน้าจอหลัก ด้านซ้ายมือจะมีการแสดง J2EE perspective
 - ในส่วนนี้จะมี title bar แสดงว่า perspective อะไรที่กำลังแสดงผลอยู่ โดยคุณอาจทำการเลือก Data, Debug, J2EE, Java และ อื่นๆ ที่สามารถเลือกเพิ่มได้
 - ถ้า title bar ไม่แสดงข้อความ J2EE ให้คุณทำการเลือก J2EE perspective จากเมนู **Window->Open Perspective->Other** และเลือก **J2EE**
 - Title bar จะแสดงข้อความ J2EE และ J2EE perspective



รูปที่ 2 แสดง J2EE perspective และ การเลือกเมนูเพื่อแสดง J2EE perspective

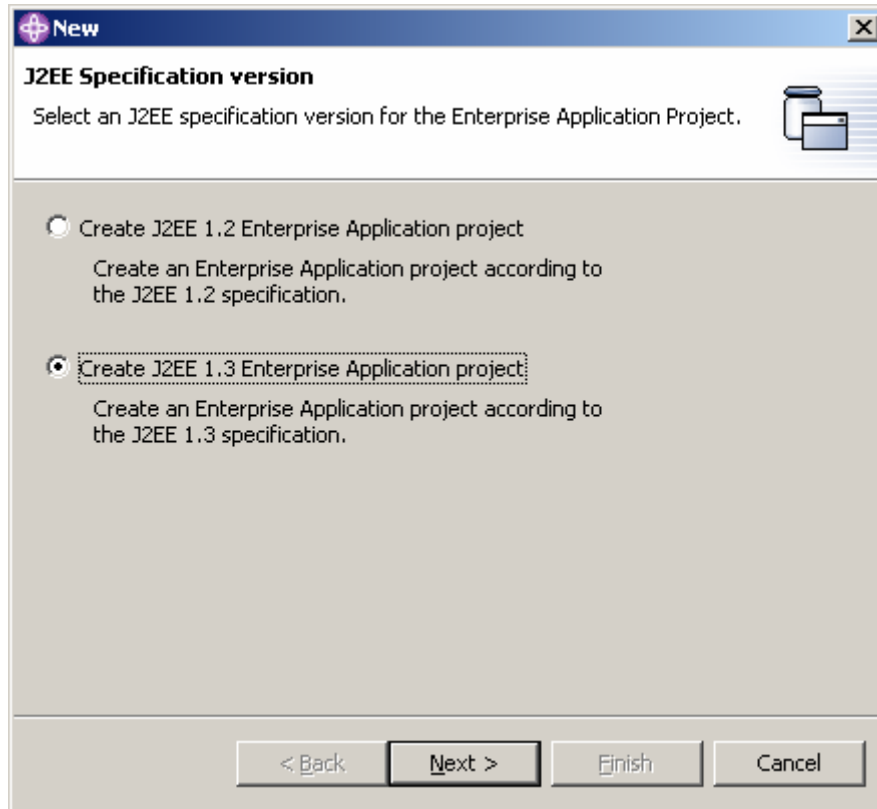
3. ที่ตำแหน่งของ J2EE perspective จะทำการสร้าง Enterprise application โดยตั้งชื่อว่า **LibraryEnterpriseApplication** ซึ่งกำหนดเลือกเป็น J2EE 1.3 Enterprise application

- ในการสร้าง Enterprise application สามารถเลือกได้ผ่านทางปุ่ม **Create an Enterprise Application** หรือคลิกเลือกจากเมนู **File -> New -> Enterprise Application Project** เพื่อเปิดหน้าต่างในการสร้าง Project ขึ้นมา



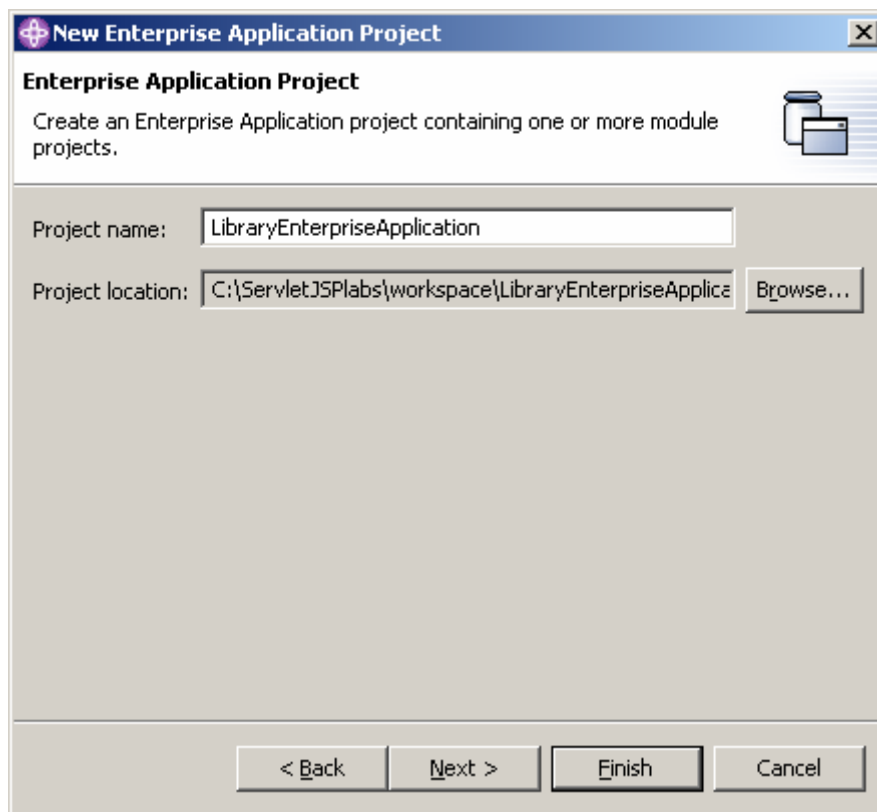
รูปที่ 3 แสดงปุ่ม Create an Enterprise Application บนทูลบาร์

- ทำการคลิกเลือก **Create J2EE 1.3 Enterprise Application Project** แล้วคลิกปุ่ม Next ดังรูป



รูปที่ 4 แสดงหน้าต่างเพื่อเลือกรุ่นของ J2EE

- ในช่องของ Project name ป้อน LibraryEnterpriseApplication

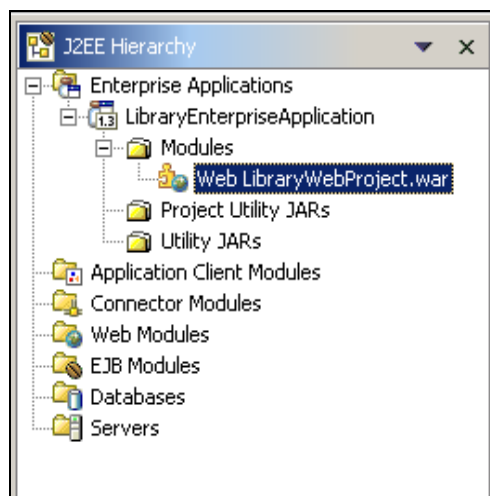


รูปที่ 5 แสดงหน้าต่างเพื่อกำหนดชื่อโปรเจก

○ คลิกปุ่ม **Finish**

4. ทำการตรวจสอบว่ามีการสร้าง J2EE 1.3 Enterprise Application ที่ชื่อว่า **LibraryEnterpriseApplication** โดยที่ยังไม่มี Application Modules, Connector Modules, Web Modules, EJB Modules, Databases หรือ Servers อยู่ภายใต้โปรเจกต์ ที่สร้างใหม่

- ในช่องของ J2EE Hierarchy ในส่วนของ J2EE perspective คุณสามารถเปิดดูใน ส่วนของรายการ **LibraryEnterpriseApplication** ซึ่งจะแสดงรูป icon ที่มีเลข 1.3 อยู่ภายใน โดยจะบ่งบอกถึงว่าเป็น J2EE 1.3



รูปที่ 6 แสดง รายการของ LibraryEnterpriseApplication

- ในส่วนของรายการภายใน Project Utility JARs และ Utility JARs จะยังไม่มี รายการย่อยอยู่ภายใน (ดังรูปที่ 6)
 - นั่นหมายถึงว่า ยังไม่การสร้าง Application Client Modules, Connection Modules, Web Modules, EJB Modules, Databases หรือ Servers ใดๆขึ้นเลย
5. ขั้นตอนต่อไป จะเป็นการสร้าง Web project โดยตั้งชื่อว่า **LibraryWebProject** โดยกำหนด ว่าจะต้องเป็น **Dynamic Web Project** และจะต้องมีความเกี่ยวข้องกับ **LibraryEnterpriseApplication** ที่ได้สร้างเอาไว้แล้ว Web project ส่วนที่จะสร้างนี้ จะต้องกำหนดให้เป็น context root ของ **Library**

Information : context root จะถูกใช้ในส่วนหนึ่งของ URL เมื่อมีการทำงานของ เซิร์ฟเว็ต ทัวๆ ไปแล้ว จะเป็นรูปแบบของการเรียกใช้ HTML, servlets และ JSP ซึ่งจะมีลักษณะดังนี้ :

`http://<hostname>/Library/<resourceName>`

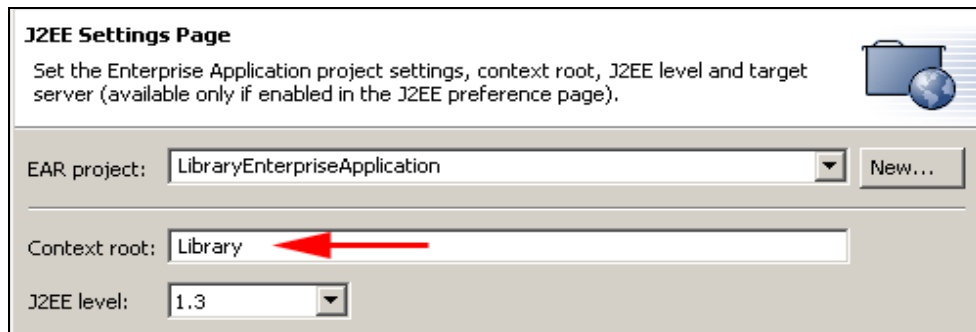
โดย hostname จะเป็นชื่อของ โฮสต์ และ <resourceName> อาจจะเป็นชื่อของ HTML, servlets หรือ JSP ที่ต้องการเรียกใช้งาน สำหรับตอนนี้ ค่า<hostname> คือ **localhost**

- ที่มุมมองของ J2EE perspective แล้วทำการเลือกเมนู **File->New->Dynamic Web Project** เพื่อเปิดหน้าต่าง **New Web Project wizard**
- ป้อน **LibraryWebProject** ลงในช่องของ Project Name และทำการคลิกเลือกที่ **Configure advanced options** คลิกปุ่ม **Next** ไปยังหน้า **J2EE Settings Page**



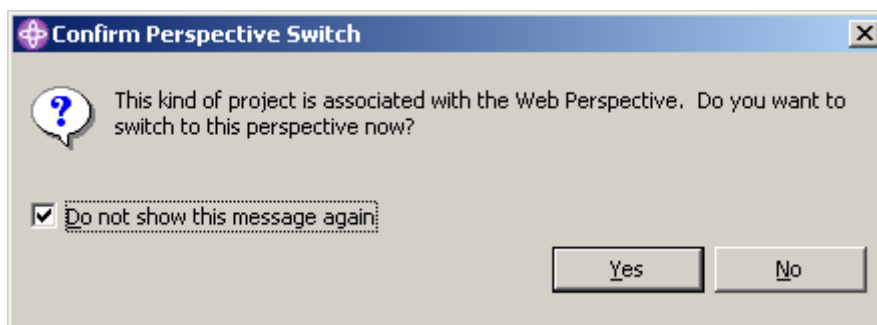
รูปที่ 7 แสดงการกำหนดชื่อ Project และเลือก Configure advance options

- ที่หน้าต่าง **J2EE Settings Page** ทำการเลือก **LibraryEnterpriseApplication** จาก **EAR project**
- เลือก **J2EE level** เป็น **1.3**
- ในช่อง Context root เปลี่ยนค่าเป็น **Library** ตรวจสอบตัวอักษรเล็กใหญ่ และการสะกดว่าถูกต้อง ก่อนจะทำขั้นตอนนี้ต่อไป



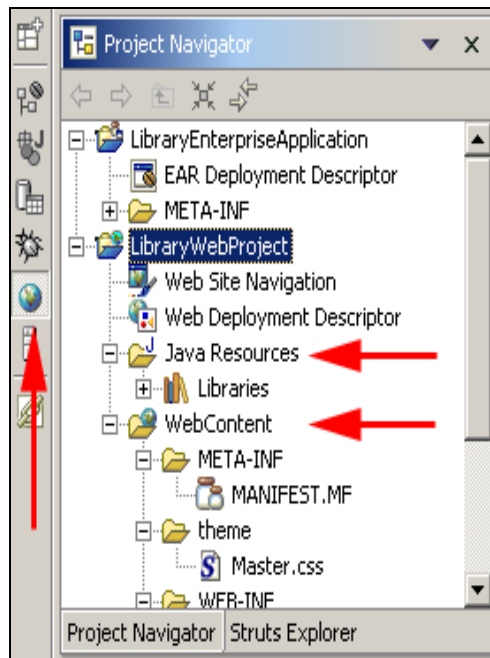
รูปที่ 8 แสดงการกำหนดชื่อ Context

- คลิกปุ่ม **Finish** เพื่อสร้าง Web project
- 6. สิ้นสุดการสร้าง LibraryWebProject ขึ้นมา จะมีหน้าต่างแสดงข้อความ เพื่อยืนยันการเปลี่ยน web perspective (มุมมองแบบเว็บ) ให้ทำการคลิกเลือก **Do not show this message again** แล้วคลิกปุ่ม **Yes** เพื่อยืนยัน มุมมองของ Web perspective มีเครื่องมือในการสร้าง servlets, JSPs และ HTML files รวมทั้งไฟล์ต่างๆที่จำเป็นในการพัฒนาเว็บ



รูปที่ 9 แสดงหน้าต่างยืนยันการเลือก web perspective

- 7. สังเกตได้ว่า LibraryWebProject จะประกอบด้วยหลายๆ ไฟล์เดอร์ **Java Resources**, **WebContent** และ **Libraries** โดย Java Resources จะเป็นที่เก็บของโค้ด servlet ส่วน WebContent จะแสดงเนื้อหาที่จะทำการเพิ่มเข้าไปใน WAR ไฟล์สำหรับ web module นี้ เมื่อคุณทำการ deploy ตัวของ enterprise application ของคุณ และสำหรับ Libraries (ที่อยู่ภายใต้ Java Resources) จะเก็บรายการของ JARs ที่ใช้สำหรับการทำงานของระบบ

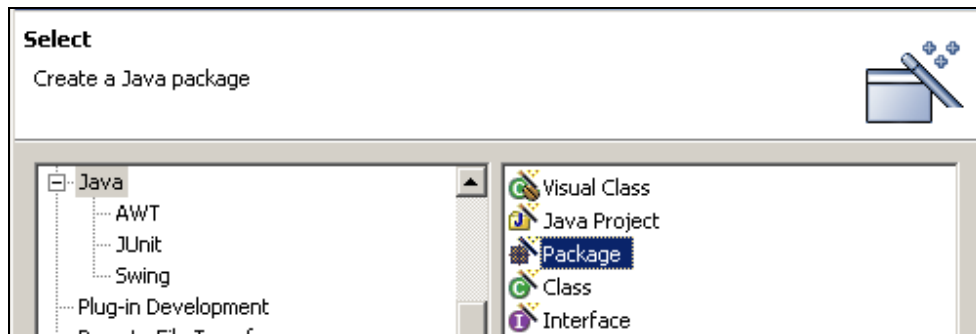


รูปที่ 10 แสดง Web perspective

ขั้นตอนที่ 2 สร้าง **Registerpatron servlet**

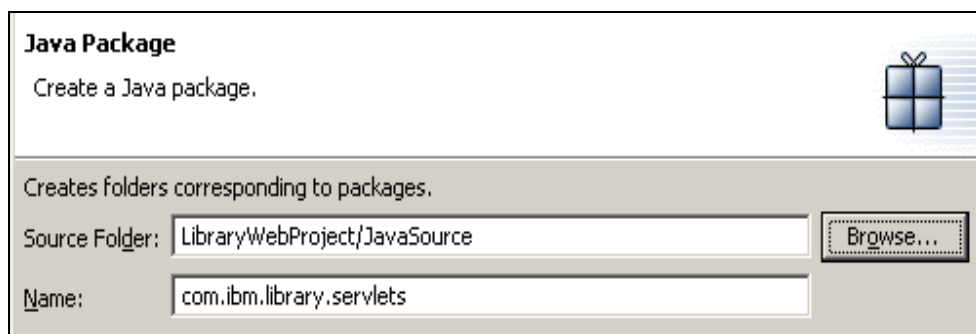
หลังจากที่ได้ทำการสร้าง enterprise application และ Web project เป็นที่เรียบร้อยแล้ว ต่อไปคุณจะสามารถเพิ่ม Web component(องค์ประกอบของเว็บ เช่น HTML,JSPs และ servlet) ต่างๆเข้าสู่ Web project ได้ (ในที่นี้จะทำการสร้าง servlet)

1. ทำการเปิดและเลือกไปที่ Web perspective
 - เลือกจากเมนู **Window->Open Perspective->Web**
2. สร้าง Java package ใหม่เป็น **com.ibm.library.servlets**
 - คุณสามารถสร้าง Java package ใหม่ได้จาก perspective ใดๆก็ได้ แต่ใน Java perspective จะมี toolbar สำหรับจัดการมาให้ด้วย สำหรับตอนนี้คุณอยู่ใน Web perspective คุณจะต้องทำการเรียกใช้งาน wizard จากเมนู **File->New->Other...** จะปรากฏหน้าต่าง **Select**
 - เลือกเมนู **Java** จากทางด้านซ้าย และเลือก **Package** จากทางด้านขวา แล้วคลิก **Next** เพื่อเปิดหน้าต่าง **Create a Java package**



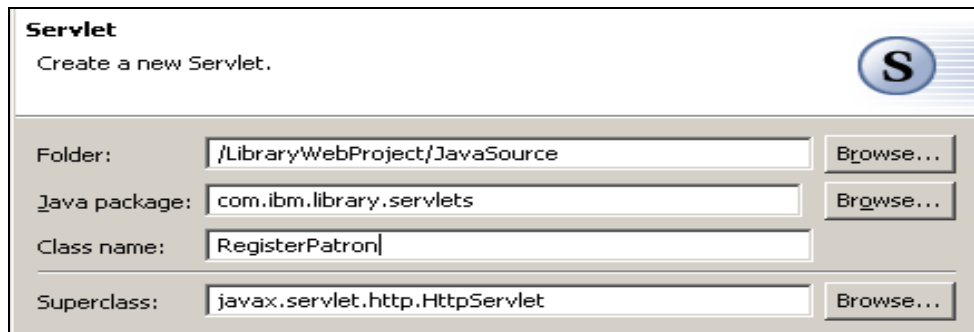
รูปที่ 11 แสดงหน้าต่าง Select ในการสร้าง Java package ใหม่

- คลิกปุ่ม **Browse** เพื่อเปิดหน้าต่าง **Folder Selection**
- ทำการคลิกเปิดเข้าไปใน LibraryWebProject และเลือก **JavaSource** คลิกปุ่ม **OK**
- ในช่อง Source Folder จะปรากฏข้อความ **LibraryWebProject/JavaSource**
- กำหนดค่าในช่อง Name เป็น **com.ibm.library.servlets** และคลิกปุ่ม **Finish** เพื่อสร้าง Java Package ใหม่



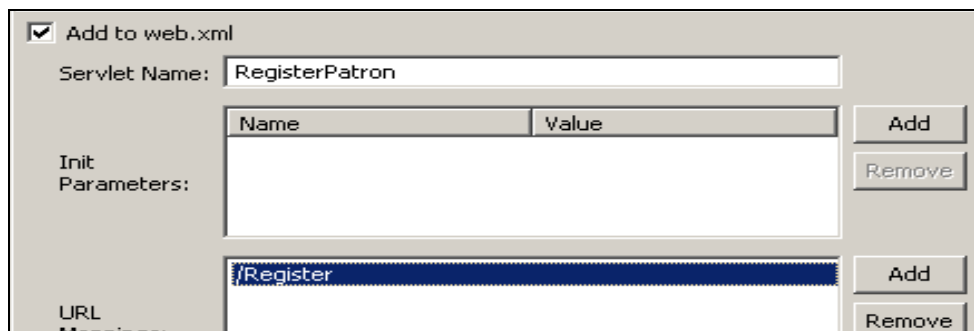
รูปที่ 12 แสดง การกำหนดที่เก็บ และการกำหนดชื่อ Java Package ที่จะสร้างใหม่

3. สร้าง servlet ที่ชื่อ **RegisterPatron** ใน package ชื่อ **com.ibm.library.servlets** โดย servlet ใหม่จะทำการ extend (สืบทอด) จากคลาส **javax.servlet.http.HttpServlet** และ implement interface คลาส **javax.servlet.Servlet** โค้ดภายในจะทำการ Override 2 เมธอด คือ **doGet()** และ **doPost()** หลักจากนั้นจะสร้าง map servlet ของคุณกับ URL pattern ตั้งชื่อเป็น **/Register**
 - จาก Web perspective ทำการเลือกเมนู **File->New->Servlet** เพื่อเปิดหน้าต่าง **New Servlet wizard**
 - ตรวจสอบค่าในช่อง Folder = **/LibraryWebProject/JavaSource** และ Java package = **com.ibm.library.servlets**
 - ในช่องของ Class name กำหนดค่าเป็น **RegisterPatron** และในช่อง Superclass จะต้องมีค่าเป็น **javax.servlet.http.HttpServlet** แล้วคลิกปุ่ม **Next**



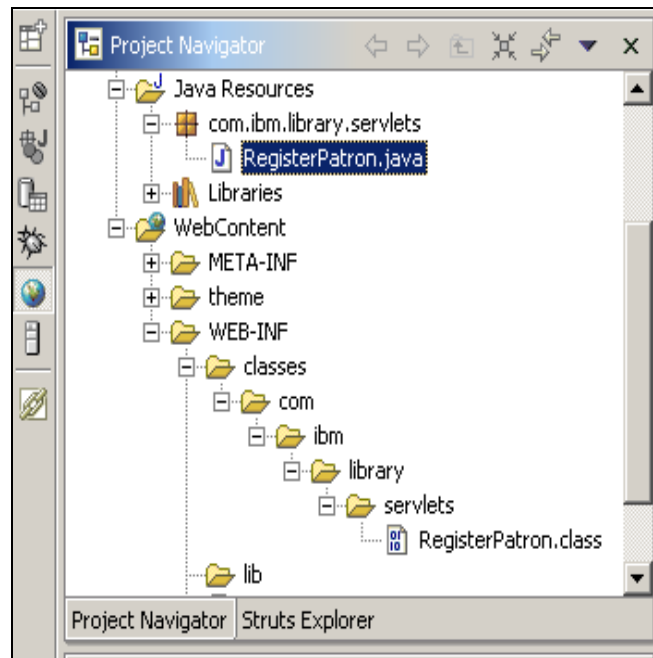
รูปที่ 13 แสดงกำหนดค่าเพื่อสร้าง RegisterPatron servlet

- ตรวจสอบค่าในช่อง Interface ให้เป็น **javax.servlet.Servlet** และมีการคลิกเลือก modifies ในส่วนของ **public** ส่วนของ method stub มีการเลือก **doGet()** และ **doPost()** สุดท้ายตรวจสอบว่า มีการคลิกเลือก **Inherited abstract methods** ไว้ จากนั้นทำการคลิกปุ่ม **Next**
- ที่หน้าต่าง Deployment descriptor ตรวจสอบว่ามีการคลิกเลือก **Add to web.xml** ไว้ โดยไฟล์ Web Deployment Descriptor นี้ จะใช้ในการ map ระหว่าง ชื่อ URL กับคลาส servlet ในที่นี้คือ คุณมี servlet ที่ชื่อ **RegisterPatron** และค่า default ของชื่อ URL จะเป็น **/RegisterPatron**
- ทำการเปลี่ยน URL mapping จาก **/RegisterPatron** ไปเป็น **/Register** จากนั้นทำการคลิกปุ่ม **Finish** เพื่อสร้าง servlet



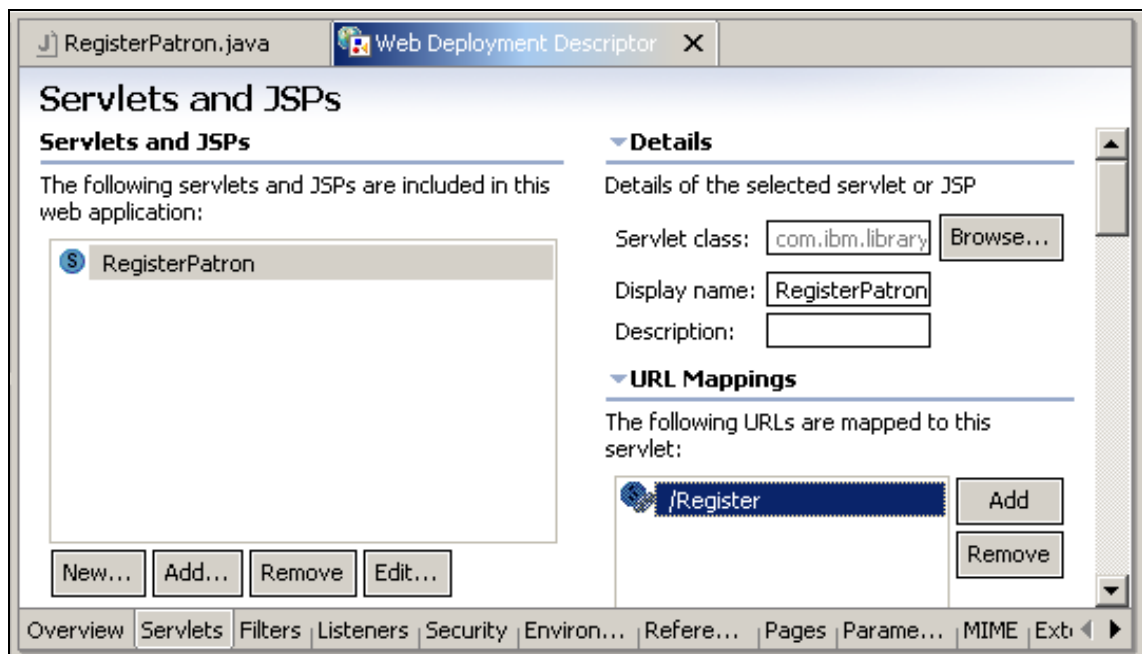
รูปที่ 14 แสดงการแก้ไข URL Mapping สำหรับ Register servlet

4. ส่วนของ edit window จะทำการแสดงคลาส **RegisterPatron** ที่เราได้ทำการสร้างขึ้นมา และทำการตรวจสอบไฟล์ **RegisterPatron.java** ว่าอยู่ใน Java Package ที่ **com.ibm.library.servlets** หรือไม่



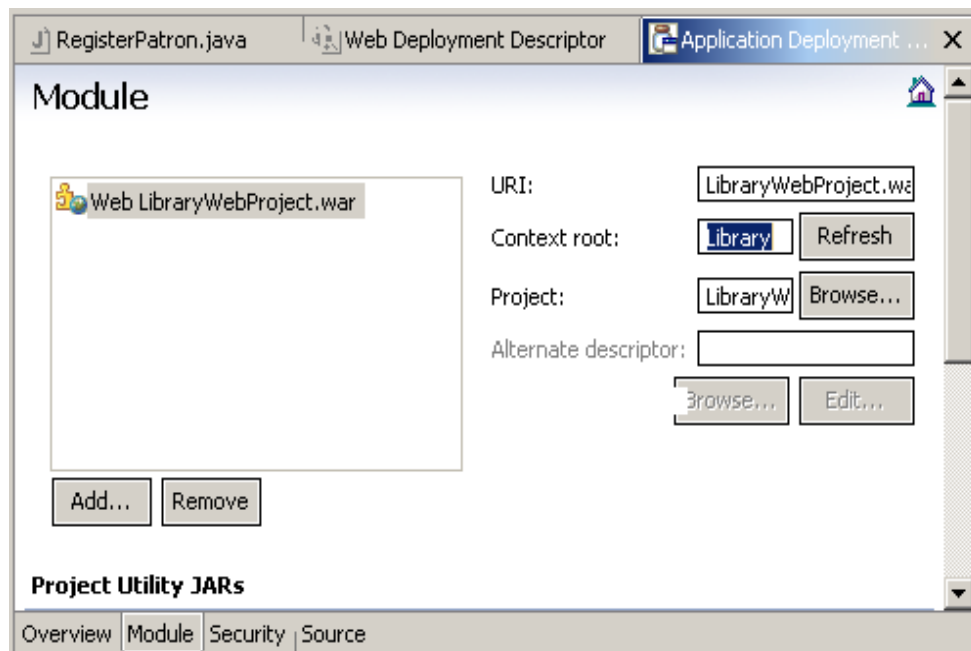
รูปที่ 15 แสดง คลาส RegisterPatron จากหน้าต่าง Web perspective

5. ทำการดับเบิลคลิกที่ **Web Deployment Descriptor** ซึ่งอยู่ใน แท็บ Project Navigator บน Web perspective จากนั้นที่หน้าต่างด้านขวา ให้คลิกแท็บ **Servlets** และเลือกที่ **RegisterPatron** servlet ทำการตรวจสอบในส่วนของ URL Mappings ว่ามีรายการของ **/Register** หรือไม่



รูปที่ 16 แสดงหน้าต่าง Web Deployment Descriptor

6. ทำการดับเบิลคลิกบน **EAR Deployment Descriptor** ที่เป็นรายการอยู่ใน **LibraryEnterpriseApplication** เพื่อแก้ไข **application.xml** โดยเลือกแท็บ **Module** และเลือก **LibraryWebProject.war** ทำการตรวจสอบค่าในช่อง **Context Root** ว่าเป็น **/Library** หรือไม่ จากนั้นให้ทำการปิดหน้าต่างนี้ออกไป



รูปที่ 17 แสดงหน้าต่าง Application Deployment Descriptor

Information : LibraryEnterpriseApplication จะแสดง EAR และจะประกอบด้วย Web Module ชื่อ LibraryWebProject ซึ่ง LibraryWebProject จะเป็น WAR ไฟล์ และมี Context root เป็น **/Library** โดยจะทำการ map การทำงานไปยัง LibraryWebProject นอกจากนั้น ภายใน LibraryWebProject จะมี URL Mapping ระหว่าง **/Register** กับ `com.library.library.servlets.RegisterPatron` โดยผู้ใช้งานจะเรียกผ่านทาง URL

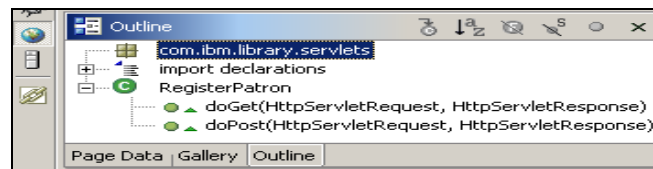
`http://<hostname>/Library/Register`

`/Library` จะเป็นค่าที่ทำให้ LibraryEnterpriseApplication ทำตามการร้องขอ ด้วย LibraryWebProject และ LibraryWebProject ได้ทำการ map ค่าของ `/Register` ไปยัง servlet ที่มีชื่อว่า `com.ibm.library.servlets.RegisterPatron` หลักจากนั้น Web container จะทำการเรียกใช้ RegisterPatron servlet

ขั้นตอนที่ 3 เพิ่มการทำงานให้กับ RegisterPatron servlet

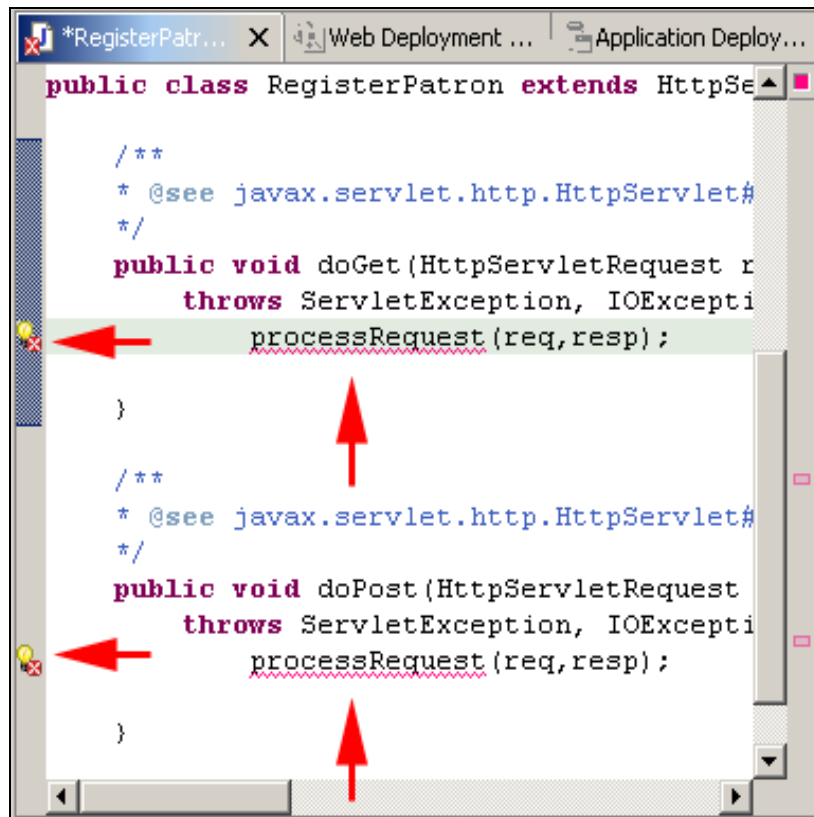
Servlet นี้จะถูกเรียกใช้งานเมื่อมีผู้ที่ต้องการเป็น สมาชิกห้องสมุด ทำการลงทะเบียน กับทางห้องสมุด โดยในขั้นตอนนี้จะเป็นขั้นตอนง่ายๆ การสร้าง id ใหม่และแสดงผลการทำงาน ว่าได้เพิ่มสมาชิกเสร็จสิ้น ให้ผู้ใช้ได้รับทราบ ซึ่งการทำงานในส่วนจริงจะอยู่ใน แบบฝึกหัดต่อไป

1. เปิดไฟล์ **RegisterPatron.java** เพื่อการแก้ไข
2. ทำการเปิด **Outline view**
 - ทำการค้นหาในส่วนต่างๆของหน้าจอ หากไม่พบส่วนหน้าต่างที่ชื่อ “Outline” ให้ทำการเลือกที่เมนู **Window->Show View->Outline** จะมีหน้าต่างปรากฏทางด้านซ้ายมือ ซึ่งภายในจะประกอบด้วยช่อง **Page Data** และ **Gallery**



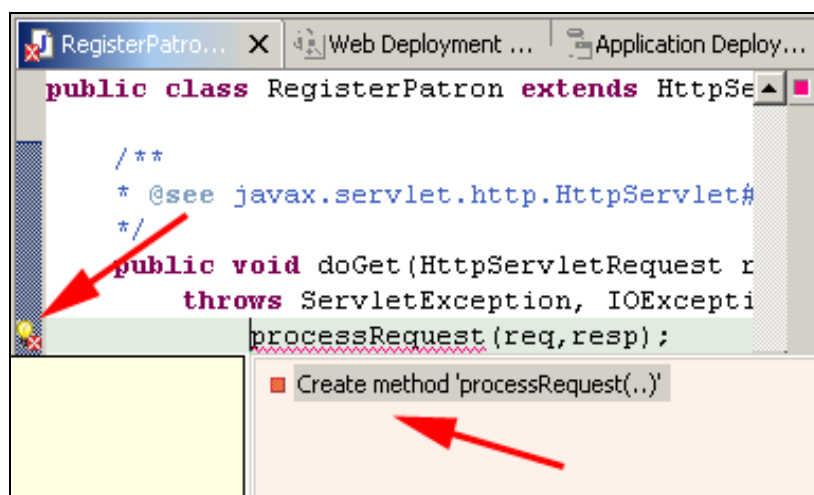
รูปที่ 18 แสดงหน้าต่าง Outline View

3. **Outline view** จะมีการทำงานที่ประสานกับช่องแก้ไขโค้ด โดยเมื่อคุณทำการเลือกไปยัง รายการเมธอด ที่อยู่ใน Outline view ทางด้านของ ช่องแก้ไขโค้ดจะทำการเลื่อนเพื่อแสดง โค้ดในส่วนเมธอด นั้นๆ ให้เห็นทันที
4. ทำการแก้ไขเมธอด doGet และ doPost ให้เรียกใช้เมธอด processRequest โดยมี req และ resp เป็นค่าอาร์กิวเมนต์ ของเมธอด จะสังเกตได้ว่า ข้อความ processRequest จะมีเส้นขีดเส้นใต้สีแดง และมีรูปหลอดไฟ แสดงให้เห็นดังรูปที่ 19



รูปที่ 19 แสดงการแก้ไขโค้ด

5. ทำการคลิกที่รูปหลอดไฟ จะปรากฏหน้าต่าง ให้ทำการดับเบิ้ลคลิกบน **Create method** ‘processRequest(..)’ (คุณสมบัตินี้เรียกว่า Quick Fix และช่วยให้การทำงานดีขึ้น)



รูปที่ 20 แสดงการคลิกเพื่อสร้างเมธอด processRequest

6. สังเกตได้ว่าในส่วนชื่อไฟล์ของโค้ด(ทางด้านบนสุด) เมื่อมีการแก้ไขโค้ดแล้วจะปรากฏข้อความว่าเป็น *RegisterPatron.java ซึ่งเครื่องหมาย * จะหายไปเมื่อคุณทำการ บันทึกการแก้ไขโค้ด

- คุณสามารถบันทึกการทำงานได้หลายวิธี อาจกดปุ่ม คีย์ **Ctrl+S** ,ผ่านทางเมนู **File->Save** ,ผ่านทางปุ่ม **Toolbar** หรือ ผ่านทางการคลิกเลือก **Save** จาก **context menu** ภายในส่วนแก้ไขโค้ด

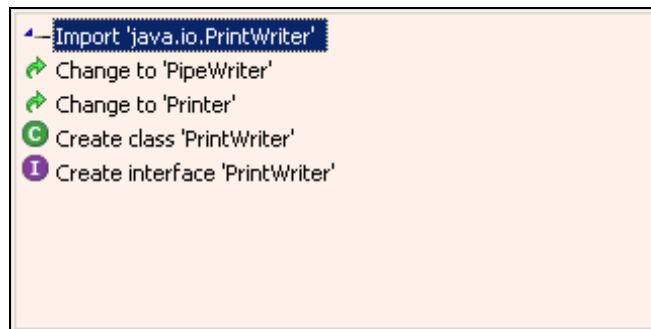
7. การแก้ไข เมธอด **processRequest** จำเป็นจะต้องใช้คลาส **PrintWriter** จากคลาส **HttpServletResponse** ซึ่งเป็นพารามิเตอร์ของเมธอดนี้ นอกจากนี้ ยังจำเป็นต้องรับค่า **id** ใหม่ผ่านทางเมธอด **LibraryIdGenerator.generatedId** โดยเมธอด **generatedId** จะเป็นแบบ **static** ซึ่งจะส่งค่าที่เป็น **int** กลับมา

Information : การทำงานของ **Quick Fix** (แสดงผ่านทาง หลอดไฟสีเหลือง) จะช่วยในการพัฒนา ถ้าคุณประกาศคลาสที่ไม่มี ประกาศคลาสผิด เรียกใช้เมธอดที่ยังไม่มีสร้างไว้ หรือข้อผิดพลาดอื่นๆ จะมีการแสดงหลอดไฟ ที่พร้อมให้คุณคลิกและเลือกแนวทางแก้ไขที่มีให้ มาให้กับคุณเพื่อเลือกวิธีการแก้ไขที่เหมาะสมต่อไป

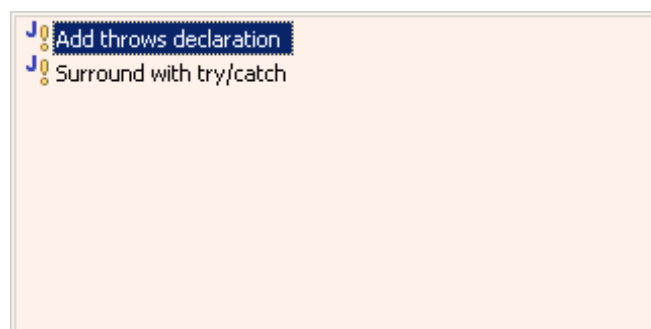
- การแก้ไข เมธอด **processRequest** โดยทำการดึงคลาส **PrintWriter** จากมาจากคลาส **response** ของ **servlet** โดยใช้เมธอด **getWriter()**

```
PrintWriter out = resp.getWriter();
```

- ใช้ **Quick Fix** ในการเพิ่มการ **import** คลาส **java.io.PrintWriter** และเพิ่มการ **throws** ให้กับเมธอด

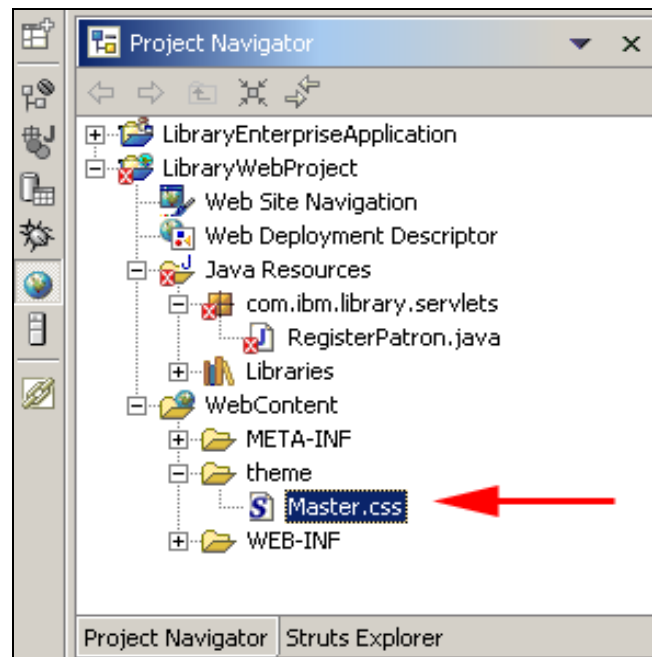


รูปที่ 21 แสดงการคลิกเพื่อเพิ่มคลาส **PrintWriter**



รูปที่ 22 แสดงการคลิกเพื่อเพิ่ม **throws** ให้กับเมธอด **processRequest**

- ขั้นตอนต่อไป จะเป็นการรับค่า id จากเมธอด **LibraryIdGenerator.generateId**
`int id = LibraryIdGenerator.generateId();`
- สุดท้ายจะเป็นการสร้างเอกสาร HTML ที่จะแสดงผลการทำงาน ให้ทำการตรวจว่าได้มีการเพิ่มไฟล์ Master.css เข้ามาดังรูปที่ 23 และทำการเพิ่มโค้ด ต่อไปนี้เข้าไป



รูปที่ 23 แสดงตำแหน่งไฟล์ Master.css

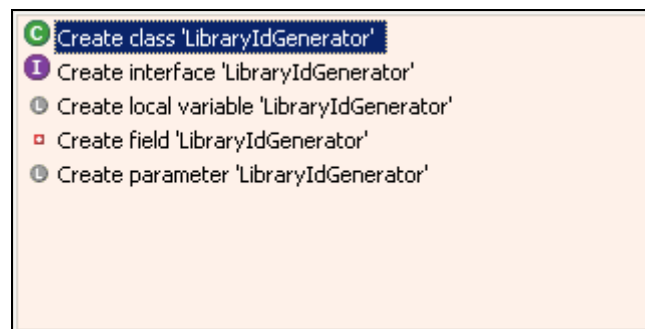
```
out.println("<HTML>");
out.println("<HEAD><TITLE>Patron added</TITLE></HEAD>");
out.println("<LINK rel=\"stylesheet\" type=\"text/css\" +
            \"href=\"/Library/theme/Master.css\">");
out.println("<BODY>");
out.println("<P>");
out.println("Patron with id "+id+" has been added.");
out.println("</BODY>");
out.println("</HTML>");
```

- ทำการบันทึกการเปลี่ยนแปลง และตรวจสอบว่ามีข้อผิดพลาดอะไรบ้าง นอกเหนือจากข้อผิดพลาดบรรทัดของ **"LibraryIdGenerator cannot resolved"**

ขั้นตอนที่ 4 สร้างคลาส **LibraryIdGenerator** และเมธอด **generateId**

ขั้นตอนต่อไป เป็นการสร้างคลาส **LibraryIdGenerator** โดยจะไปสร้างไว้ใน package ของ **com.ibm.library.servlets** โดยคลาสจะประกอบด้วยสแตติกเมธอดเพียง เมธอดเดียว ที่ชื่อว่า **generateId** ซึ่งจะส่งค่าของ **id** ของสมาชิกห้องสมุดกลับมา

1. ทำการสร้างคลาส **LibraryIdGenerator** โดยจะไปสร้างไว้ใน package ของ **com.ibm.library.servlets** ซึ่งสามารถใช้ Quick Fix ในการสร้างได้
 - ทำการคลิกที่รูปหลอดไฟ (Quick Fix) ที่อยู่ในหน้าบรรทัดแรกของ เมธอด **LibraryIdGenerator.generateId** ภายในไฟล์ **RegisterPatron.java**
 - ดับเบิ้ลคลิกที่ **Create class 'LibraryIdGenerator'** จะปรากฏหน้าต่าง Java Class wizard ทำการตรวจสอบค่าต่างๆ และคลิก **Finish** คลาส **LibraryIdGenerator** จะถูกสร้างและจะแสดงในช่องแก้ไขโค้ด ไฟล์ **LibraryIdGenerator.java**



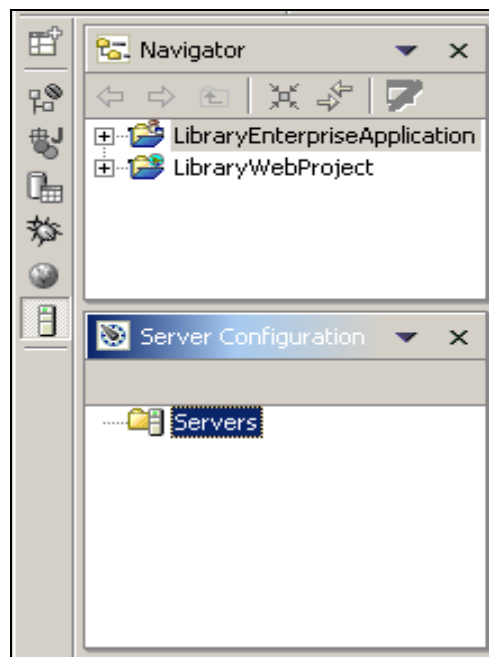
รูปที่ 24 แสดง การเลือกรายการเพื่อสร้างเมธอดใหม่

2. ทำการสร้างเมธอด **public static generateId** ภายในคลาสของ **LibraryIdGenerator** และเมธอด **generateId** จะส่งค่า **int** กลับมา แต่ในขั้นตอนนี้จะส่ง ค่า 0 กลับมาเพียงเพื่อต้องการตรวจสอบการทำงาน โดยรวมของทั้งหมดก่อน
 - กลับไปเปิดไฟล์คลาส **RegisterPatron** แล้วตรวจสอบข้อผิดพลาด ว่ายังเป็นข้อผิดพลาดที่แจ้งว่า เมธอด **generateId** ยังไม่ถูกสร้างขึ้น
 - ใช้ Quick Fix ในการสร้างเมธอด **generateId** ภายในคลาส **LibraryIdGenerator**
 - คุณจะถูกกำหนดให้กลับไปเปิดหน้าต่างแก้ไขโค้ด **LibraryIdGenerator.java**
 - ทำการตรวจสอบว่าเมธอดส่งค่า 0 กลับหรือไม่ (return 0)
 - ทำการบรรทัดที่คลาส **LibraryIdGenerator** จะต้องมีรายการข้อผิดพลาดใน Tasks view

ขั้นตอนที่ 5 สร้าง **Server Project** สำหรับการทดสอบงาน

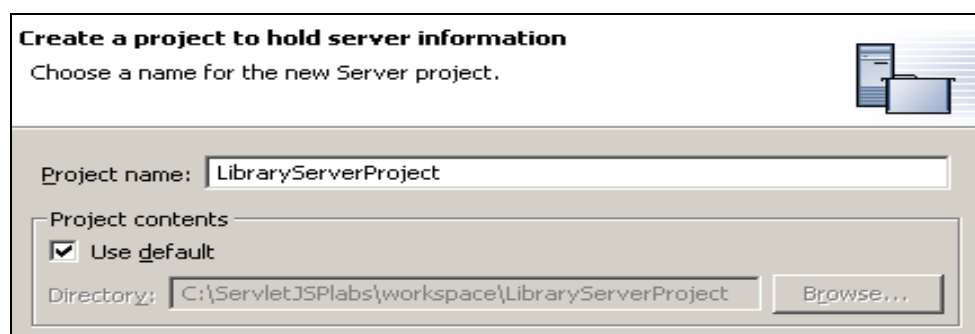
ในการเรียกใช้งาน servlet คุณจำเป็นต้องสร้าง และ ปรับแต่งค่า ของ **Server Project** โดยมีวิธีการสร้างที่เร็ว แต่หากว่าคุณต้องการควบคุมการทำงานที่มีความซับซ้อนของ ค่าการปรับแต่ง **Server** แล้วละก็วิธีนี้จะไม่เหมาะกับงานคุณ

1. ทำการเปิดหน้า **Server perspective** และสร้าง **Server Project** ใหม่ โดยตั้งชื่อเป็น **LibraryServerProject**
 - เลือกจากเมนู **Window->Open Perspective->Server** เพื่อเปลี่ยนไปยัง **Server perspective**



รูปที่ 25 แสดงหน้าต่าง Server Perspective

- เลือกที่จะคลิกปุ่ม **Create Server Project** จาก toolbar หรือ คลิกจากเมนู **File->New->Server Project** เพื่อเปิดหน้าต่าง **Create a New Server Project**
- กำหนดชื่อของโปรเจกเป็น **LibraryServerProject** และคลิกปุ่ม **Finish** เพื่อสร้าง

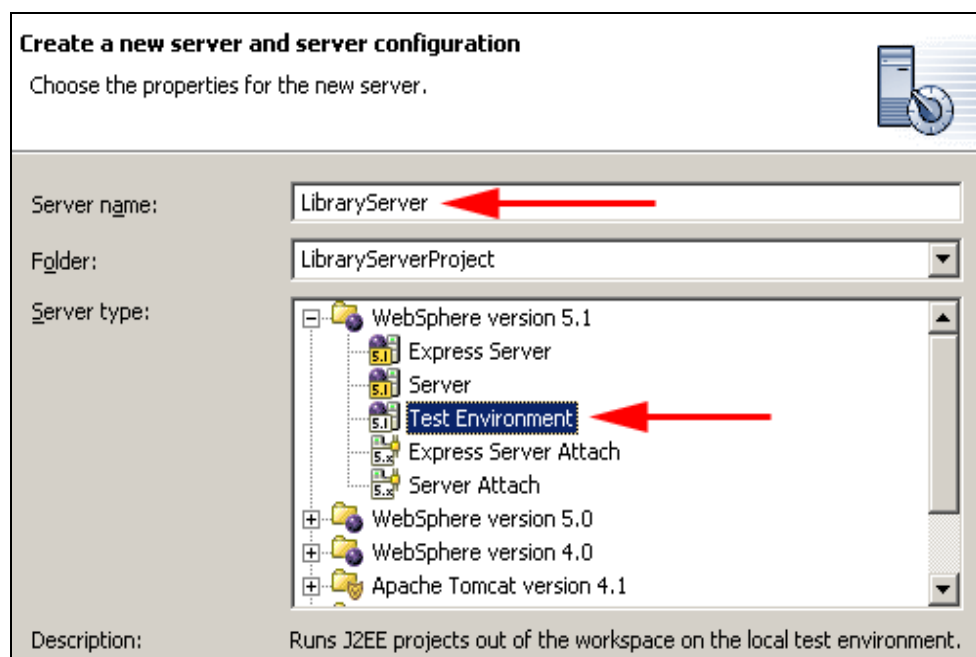


รูปที่ 26 แสดงหน้าต่างการสร้าง Server Project

2. สร้าง Server และ กำหนดค่าปรับแต่งของ Server ใหม่ และกำหนดชื่อเป็น **LibraryServer** ซึ่งจะเป็น **WebSphere Version 5.1 Test Environment** ในการสร้าง

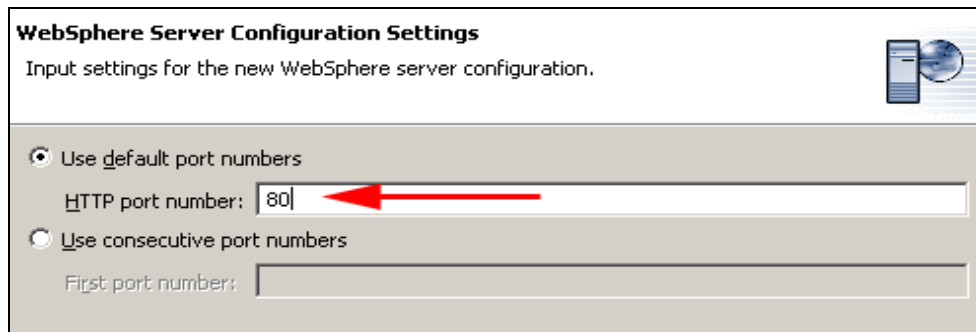
LibraryServer จะต้องสร้างไว้ภายใน **Library ServerProject** และกำหนดให้ **LibraryServer** ใช้งาน พอร์ตหมายเลข **80**

- คุณสามารถคลิกเลือกที่ปุ่ม **Create Server and Server Configuration** จาก toolbar หรือเลือกจากเมนู **File->New->Server and Server Configuration** จะมีการแสดงหน้าต่าง **Create a New Server and Server Configuration** ออกมา
- กำหนดชื่อเซิร์ฟเวอร์เป็น **LibraryServer** และตรวจสอบว่า ค่าของ Folder เป็น **LibraryServerProject**
- ทำการคลิกเปิดรายการของ **WebSphere version 5.1**
- เลือกที่ **Test Environment**
- คลิกปุ่ม **Next**



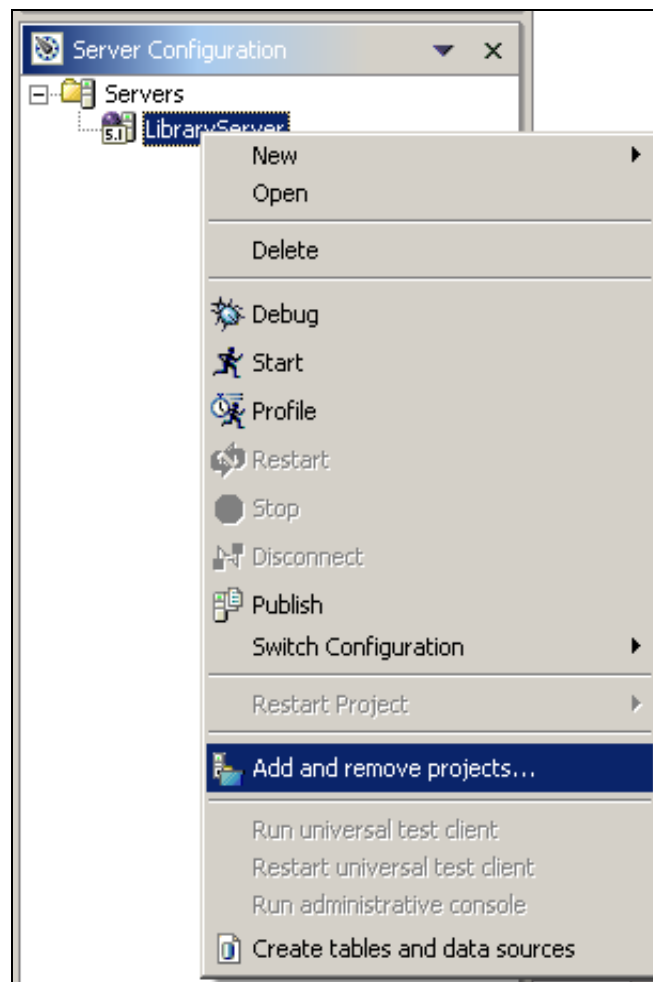
รูปที่ 27 แสดงการ สร้างโปรเจค Server เพื่อทดสอบการทำงาน

- เปลี่ยนค่าของ **HTTP port number** เป็น **80** และคลิกปุ่ม **Finish** เพื่อสร้าง Server และ Server configuration



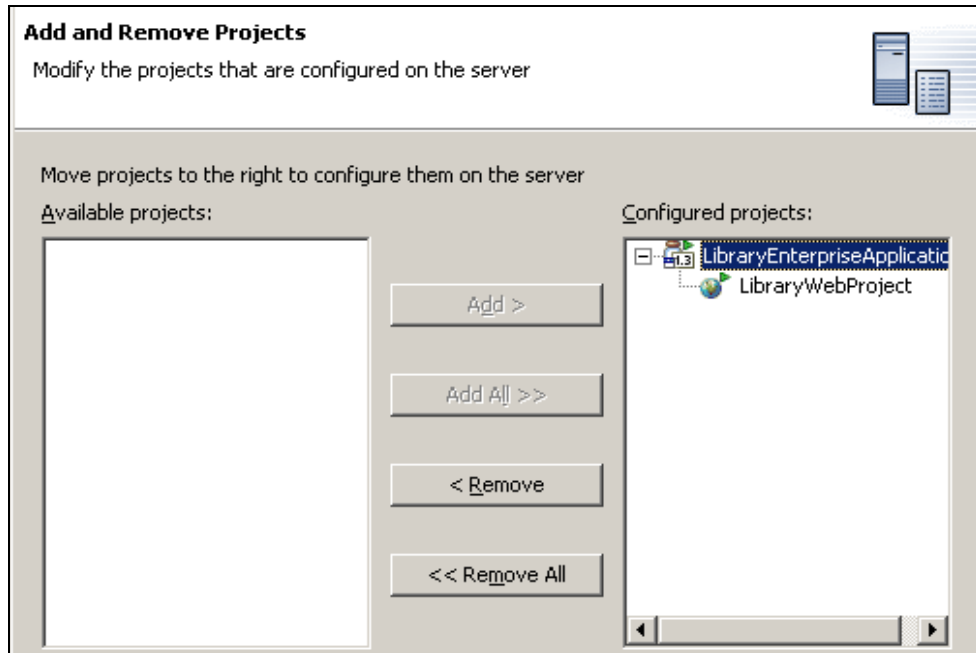
รูปที่ 28 แสดงการกำหนดหมายเลข port ของ Server

3. ขั้นตอนแก้ไขเพื่อปรับแต่งเพื่อให้ **LibraryServerProject** มีการติดต่อทำงานร่วมกับ **LibraryEnterpriseApplication** ผ่านทางการกำหนดค่าของ **LibraryServer**
 - เปิดไปที่หน้าต่าง **Server Configuration** และคลิกเปิดรายการต่างๆของ **Server**
 - คลิกเลือกที่ **LibraryServer** และคลิกเมาส์ขวา

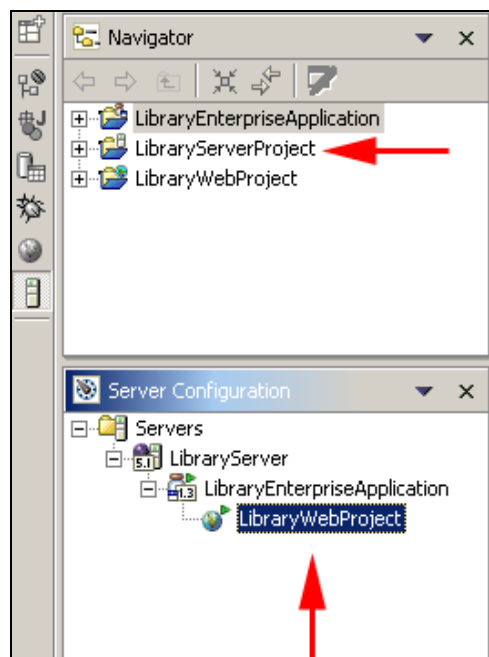


รูปที่ 29 แสดงการเพิ่ม project เข้าสู่ Server

- เลือก **Add and remove project** จากเมนู pop-up จะปรากฏหน้าต่าง Add and Remove Projects และทำการเลือก **LibraryEnterpriseApplication** ที่อยู่ด้านซ้ายมือ จากนั้นคลิกปุ่ม **Add** เพื่อทำการเพิ่มโปรเจกต์ไปยังด้านขวา แล้วคลิกปุ่ม **Finish**



รูปที่ 30 แสดงการเพิ่ม Project ที่ต้องการทดสอบเข้าสู่ Server



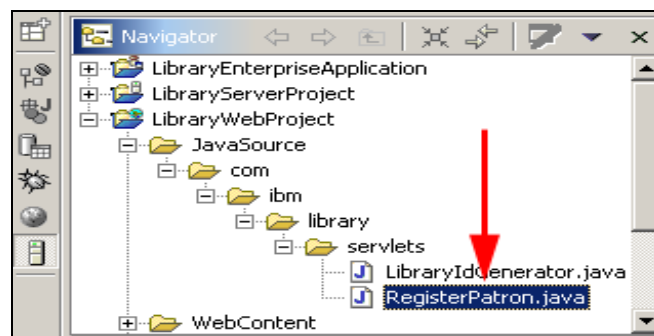
รูปที่ 31 แสดง Project ที่ถูกเพิ่มเข้ามาเพื่อทดสอบ

ขั้นตอนที่ 6 การทดสอบ servlet

Servlet จะถูกเรียกผ่านทาง เว็บเบราว์เซอร์โดยมี URL ที่มีการ map เอาไว้ ซึ่งขณะนี้ hostname คือคำว่า “localhost”, context root คือ Library และมี mapping name เป็นคำว่า “Register” เมื่อต้องการสั่งทำงานจะผ่านทาง URL:

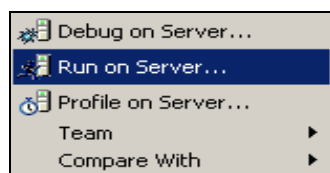
<http://localhost/Library/Register>

1. ทำการเปิด **WebSphere Test Environment** จากหน้าต่างของ Server perspective เพื่อสั่งให้ RegisterPatron servlet ทำงาน
 - ทำการเลือกไปที่หน้าต่าง Server perspective
 - ค้นหาคลาส **RegisterPatron** จาก LibraryWebProject จาก com.ibm.library.servlets และทำการคลิกเลือก



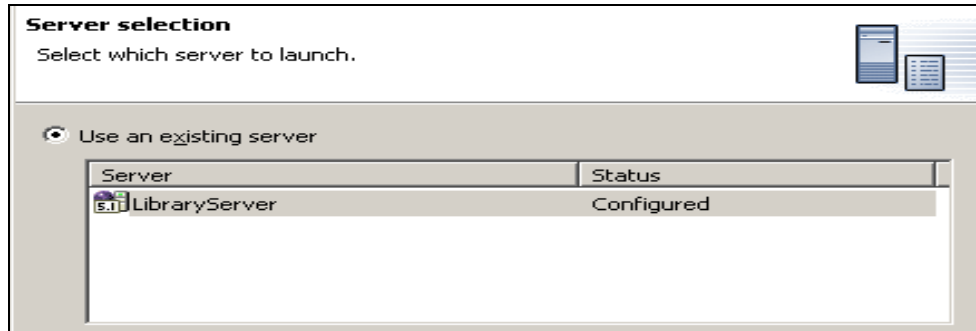
รูปที่ 32 แสดงการเลือกเพื่อตั้ง run servlet

- คลิกเมาส์ขวา และเลือก **Run on Server** จากเมนู context จะปรากฏหน้าต่าง **Server Selection**



รูปที่ 33 แสดงการเลือกรายการ **Run on Serve...**

- เลือกรายการ **LibraryServer** จากรายการของ **Use an existing server**
- ตรวจสอบรายการ **Set Server as project default (do not prompt)** ว่าถูกเลือกอยู่
- คลิกปุ่ม **Finish** เพื่อสั่งให้ Servlet ทำงาน



รูปที่ 34 แสดงการเลือกรายการ Server

เมื่อทำการเริ่มต้นการทำงานของ WebSphere Test Environment โดยที่ยังไม่มีการเรียกใช้งานมาก่อนหน้านี้

- 1) โปรแกรมจะถูกส่งผ่านเข้าสู่ server
- 2) Server (WebSphere Application Server) เริ่มต้นการทำงาน
- 3) ปรากฏหน้าต่างเว็บเบราว์เซอร์
- 4) เว็บเบราว์เซอร์ทำการเรียกใช้งานผ่าน URL ของ servlet
- 5) Servlet จะถูกเรียกให้ทำงาน และจะเริ่มต้นทำงาน
- 6) Servlet แสดงผลการทำงานที่หน้าต่างเว็บเบราว์เซอร์

Information : ถ้าคุณไม่สามารถสั่ง run servlet ได้ ให้ทำการตรวจสอบ เว็บเบราว์เซอร์ว่าไม่มีการกำหนดใช้ proxy ไว้หรือไม่ นอกจากนี้ให้ทำการตรวจสอบว่า มีการทำงานของ Web Server อยู่บนเครื่องหรือไม่

2. จะสังเกตได้ว่าข้อความที่แสดงบนเว็บเบราว์เซอร์ มีค่า id เป็น 0 ซึ่งได้จาก เมธอด **LibraryIdGenerator.generatedId** ที่ได้สร้างขึ้น
3. ทำการหยุดการทำงานของ **LibraryServer**
 - o ที่หน้าต่าง Server perspective ทำการคลิกเลือกเปลี่ยนจาก **Console view** ไปเป็น **Server view**
 - o คลิกเลือก **LibraryServer** จากรายการที่แสดง จากนั้นให้คลิกปุ่ม **Stop the server** จากปุ่ม ทูลบาร์



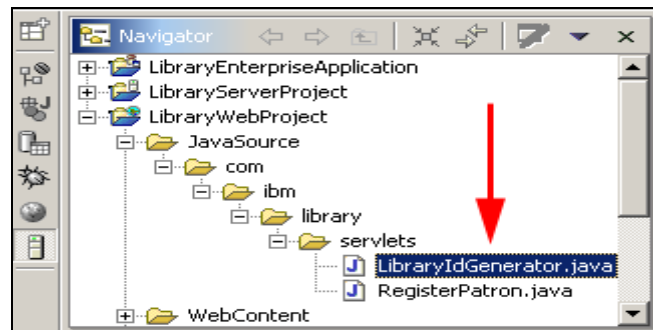
รูปที่ 35 แสดงการหยุดการทำงานของ Library Server

ขั้นตอนที่ 7 ปรับปรุงเมธอด **LibraryIdGenerator.generateId()**

ถึงขั้นตอนนี้ คุณสามารถสั่ง run servlet ได้ และต่อไปคุณจะต้องทำการสร้าง id ค่าใหม่สำหรับผู้ที่มาลงทะเบียนของห้องสมุด

1. ทำการแก้ไขเมธอด **LibraryIdGenerator.generateId** เพื่อให้สร้างค่าตัวเลขจำนวนเต็มบวก ที่มากกว่า 0 ซึ่งในตัวอย่างในแบบฝึกหัดถัดไป คุณจะนำค่านี้มาจาก ฐานข้อมูล แต่สำหรับตอนนี้ เราจะทำแค่เพียงการสุ่มค่า ขึ้นมาค่านี้
 - ทำการแก้ไขคลาส โดยเปิดไฟล์ **LibraryIdGenerator.java** และเปลี่ยนแปลงค่า return ของเมธอด **generateId** ตามนี้

```
return (new Double((Math.floor(1000000*Math.random())))).intValue();
```



รูปที่ 36 แสดงหน้าต่างของ Navigator view แสดงไฟล์ LibraryIdGenerator.java

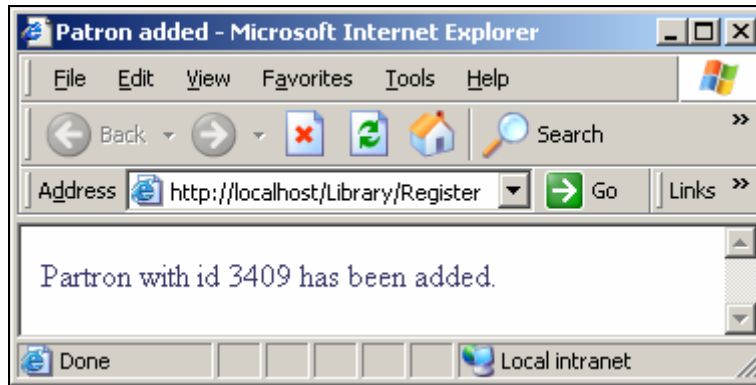
- บันทึกไฟล์ จะต้องไม่มีข้อผิดพลาดในโค้ดที่แก้ไขลงไป

ขั้นตอนที่ 8 ทำการทดสอบ **RegisterPatron** กับเมธอด **LibraryIdGenerator.generateId()** ที่ได้ปรับปรุง

1. ตรวจสอบว่าได้เปิดหน้าต่าง Server perspective ไว้แล้วทำการ run RegisterPatron
 - คลิกเลือก **RegisterPatron** จาก **Navigator** view แล้วคลิกขวา เลือก **Run on Server** จากเมนู pop up
 - Server จะถูกเรียกให้เริ่มดำเนินงาน และแสดงค่าที่ได้บนเว็บเบราว์เซอร์
2. ค่าที่ได้ใหม่จะต้องไม่เท่ากับ 0 แต่จะเป็นจำนวนเต็มบวกที่ได้จากการสุ่ม
3. ที่หน้าต่าง **Web Browser** view ถ้าคุณทำการคลิกปุ่ม **Refresh the current page** คุณก็จะได้ค่า id ใหม่แสดงออกมา (มีโอกาสน้อยมากที่จะมีค่าซ้ำกัน)
4. ทดลองเรียกใช้งาน servlet ผ่านทางเว็บเบราว์เซอร์ โดยมี URL เป็น

```
http://localhost/Library/Register
```

 - เปิดโปรแกรม Internet Explorer และป้อน URL ของ Servlet เข้าไป
 - หากไม่ได้ติดตั้งโปรแกรม Internet Explorer เอาไว้ ให้ทำการเรียกใช้ Web Browser ตัวอื่น และป้อน URL เข้าไป



รูปที่ 37 แสดงผลที่ได้เมื่อเรียก servlet ผ่านทาง Internet Explorer

5. หยุดการทำงานของ Server และปิดหน้าต่างแก้ไข และเว็บเบราว์เซอร์
6. สิ้นสุดกระบวนการสร้าง servlet ตัวแรกของคุณ

END OF LAB