

Servlet Beans

Vittayasak Rujivorakul

Lecture 9

Servlet Beans

Vittayasak Rujivorakul

Lecture 9

อะไรคือ Java Bean

- Bean สามารถนำกลับมาใช้ใหม่ได้ เป็น platform - independent และเป็น component model
- เขียนด้วยภาษา Java
- สามารถนำมาเป็นส่วนประกอบของ software ได้
- นำมาใช้ร่วมกับ application หรือ applet

Functionality of Beans

- **Introspectable** : Bean สามารถส่งไปเก็บไว้ยัง Bean properties ได้
- **Event aware** : มีส่วนของ event ที่ใช้ในการติดต่อกับ Bean ตัวอื่น
- **Customizable** : Bean สามารถ ปรับแต่งได้โดยอาจใช้ GUI editors ในการ configure Bean ได้
- **Persistent** : Bean สามารถ serialize ตัวเองและบันทึกให้คงอยู่ตลอดไปได้ โดยการ save เป็น .ser file

อะไรคือ Servlet Bean

การสร้าง Bean จำเป็นต้องเป็นไปตาม Java Beans design patterns คือ

- การ access variable จะทำผ่าน **get** และ **set** Assessor
- สามารถมี Bean properties เพื่อเก็บตัวแปรหรือสถานะ และสามารถเรียกขึ้นมาได้โดยผ่าน **get** และ **set** Assessor เช่นกัน

Servlet สามารถเลือกใช้ Bean อื่นได้ หรือ Servlet สามารถทำตัวเป็น Bean ได้

Example Servlet Bean

```
import javax.servlet.http.*;
import java.io.PrintWriter;
import java.io.IOException;
public class SampleBean extends HttpServlet {
    String stringProperty = "Hello World";
    int intProperty = 0;
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws IOException {
        // Set response headers:
        res.setContentType("text/html");
        res.setHeader("Cache-Control", "no-cache");
        // Emit current properties setting:
        ServletOutputStream out = res.getOutputStream();
        out.println("<html><head><title>SampleBean</title></head><body>");
        out.println("<p>SampleBean current configuration:</p>");
        out.println("<table border=1>");
        out.println("<tr><th align='center'>Property</th>");
        out.println("<th align='center'>Value</th>");
        out.println("<tr><td>stringProperty</td>"+getStringProperty());
        out.println("<tr><td>intProperty</td>"+getIntProperty());
        out.println("</table>");
        out.println("<p>Go to the admin to change these bean properties !");
        out.println("</body></html>");
    }
}
```

Example Servlet Bean

```
public String getStringProperty() {
    return stringProperty;
}

public void setStringProperty(String stringProperty) {
    this.stringProperty = stringProperty;
}

public int getIntProperty() {
    return intProperty;
}

public void setIntProperty(int intProperty) {
    this.intProperty = intProperty;
}
}
```

Setting Header Information

- **Res.set Header** : ใช้ในการเพิ่ม field ลงในส่วนของ header ที่ response กลับไปโดยระบุเป็นชื่อ และ value
- การควบคุม การ cache ของ browser เราใช้
 - **Cache - Control** เป็นชื่อของ field
 - **no - cache** เป็น attribute บอกกับ browser ว่าไม่ให้มีการ cache ข้อมูล เพื่อให้ข้อมูล update เสมอ

Get and Set Accessor

- เราใช้ get และ set method ในการเข้าถึงข้อมูลใน Servlet properties ซึ่งโดยมากเรามัก ตั้งค่าเหล่านี้ด้วย object Servlet Config ซึ่งอยู่ใน init method (ค่าต่าง ๆ ใน properties จะถูกเก็บไว้ที่ web server)
- ตัวอย่างเช่น

```
public void set String Property (String string Property)

public String get String Property
```

Servlet Bean Persistence

- การบันทึก Servlet Beans สามารถทำได้โดยการ Implement `java.io.Serializable` หรือ `java.io.Externalizable`
- การบันทึก ทำโดยการเขียน ผ่าน serialization file (.ser)
- เราสามารถใส่ขั้นตอนในการบันทึกนี้ไว้ในส่วนของ method `destroy`
- Java Web Server จะทำการสร้าง Servlet Bean ขึ้นมาใหม่ โดยอัตโนมัติ
- Java Web Server จะทำการสร้างไฟล์ .ser สำหรับ class bean นั้นใน แล้วบรรจุลงใน jar file

การเขียน Serialization File

```
try{
    String serFile = "/opt/JavaWebServer/servlet/SimpleBean.ser";
    ObjectOutputStream os = new ObjectOutputStream(new FileOutputStream(
        serFile));
    os.writeObject(this);
    os.flush();
    os.close();
}catch(Exception e){
    log("destroy: Exception caught");
}
```

Servlet Bean Instantiation

- Class loader จะทำการ **de serializes servlet Beans files** จาก .ser ซึ่งอยู่ใน JAR file หรือไม่อยู่ที่ใด เมื่อมีการเปลี่ยนแปลงที่ file นั้น class loader จะทำการ reload โดยอัตโนมัติ
- เราสามารถเรียกใช้ Servlet Bean ได้โดยผ่าน `server-root\ servlets ; server-root\servletBeans` หรือ Directory อื่น ที่สามารถอ้างอิงได้ใน CLASS PATH
- Bean File ของเราจะเป็น file .class, .ser และ .jar และจะต้องอยู่ใน CLASS PATH ของ server เท่านั้น ตัว web server จึงจะทำการ reload ให้เมื่อเกิดการเปลี่ยนแปลง

Invoking Servlet Beans

```
String greeting = "Hello World";
public String getGreeting(){
    return greeting;
}
public void setGreeting(String g){
    greeting = g;
}
public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException{
    PrintWriter out;
    res.setContentType("text/html");
    out = res.getWriter();

    out.println("<html>");
    out.println("<head><title> Greeting </title></head>");
    out.println("<body>");
    out.println("<h1>" + getGreeting() + "</h1>");
    out.println("</body></html>");
}
```