Advanced Internet Technology Lab

Lab # 4

Servlet Coding Basics

Eng. Haneen El-masry

March, 2014
Objectives

- To be familiar with servlet Coding.
- To create web applications by Eclipse.
- To be familiar with servlets that generate plain text and HTML.

Servlets Basic Structure

```java
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class MyServlet extends HttpServlet {

    public MyServlet() {
        super();
    }

    public void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("ResponseType");
        PrintWriter out = response.getWriter();
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        doGet(request, response);
    }
}
```

Notes:

1- Servlet must extend HttpServlet.
2- Servlet must have public no argument constructor.
3- Override doGet and/or doPost depending on whether the data is being sent by GET, POST.
These methods take two arguments:

- HttpServletRequest: lets you get at all of the incoming data.
- HttpServletResponse: lets you specify outgoing information and obtain a PrintWriter that is used to send document content back to the web browser.

doGet and doPost throw two exceptions. You must include them in the method declaration.

- ServletException.
- IOException.

4- Finally, you must import the following packages:

- java.io (for PrintWriter, IOException etc.).
- javax.servlet (for ServletException).
- javax.servlet.http (for HttpServletRequest, HttpServletResponse and HttpServlet).

**How To Generate Dynamic Web Application in Eclipse?**

1- Create a Dynamic Web Project.

File >> New >> Project >> Web >> Dynamic Web Project.
2- Insert the project name >> Next >> Next >> Finish.
Ensure that Target runtime is Apache Tomcat v7.0
3- Java classes will be in Java Resources, HTML, CSS and JSP will be in WebContent, also web.xml will be in WEB-INF.

4- To make a Packaged Servlet, create a package.
   R-Click on src >> New >> Package >> insert Package name >> Finish.
5- Create a servlet.
   R-Click on the new Package >> New >> Servlet >> insert Servlet class name >> Next.

6- Enter the Servlet Deployment Descriptor information: Initialization parameters and URL pattern >> Next.
This specifications are equivalent to the next description in web.xml:

```xml
<servlet>
    <servlet-name>PlainText</servlet-name>
    <servlet-class>Servlets.PlainText</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>PlainText</servlet-name>
    <url-pattern>/PlainText</url-pattern>
</servlet-mapping>
```

The URL of the Servlet will be: [http://localhost/Labs/PlainText](http://localhost/Labs/PlainText)

7- Finish.
8- Your Servlet is Ready 😊. Edit it.

9- Add the project to Server to run it.
   R-Click on Server >> Add and Remove >> choose your project >> Finish.
10- Start the Server or Restart it if it is already started.

11- Run the Servlet, either from Eclipse or by entering its URL in your browser.
   ➢ To run the servlet from Eclipse:
     Run >> Run As >> Run on Server >> Finish.
To run from browser enter the URL of the Servlet. If you follow the previous steps, the URL will be:

http://localhost/Labs/PlainText

Example 1: A servlet that generates Plain Text

Response content type is: text/plain
Example 2: A servlet that generates HTML

Response content type is: text/html

```java
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    // TODO Auto-generated method stub
    response.setContentType("text/html");
    PrintWriter out=response.getWriter();
    out.println("<html><head><title>WELCOME TO JSP LAB</title>" +
            "</head><body><h1>WELCOME</h1></body></html>");
}
```