Desarrollo de Aplicaciones en Red

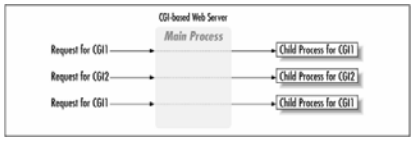
José Rafael Rojano Cáceres
http://www.uv.mx/rrojano

A few more words about

Common Gateway Interface

CGI

- *So originally CGI purpose was to let communicate a server with external applications.*
- *For that reason the CGI's life cycle was like this*



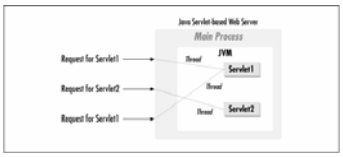
CGI

- *A improvement of CGI's life cycle was FastCGI which different is to keep a persistent process to handle the request to the same CGI.*
- *Using apache there is the option to use mod_perl, which is embedded in the daemon httpd, this let to script run faster.*

Servlet & JSP

What are servlets?

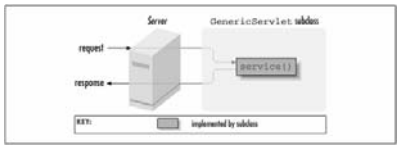
- ***A servlet is a generic server extension, a class that can be loaded dynamically to expand server functionality.***
- ***Servlets' life cycle are handled by separate threads***

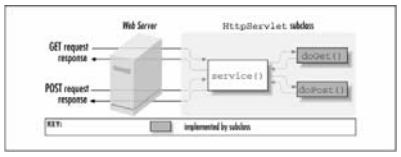


Using Servlets

- ***Sevlets use classes and interface from javax.servlet and javax.servlet.http***
  - The first contains classes and interfaces that describe and define the contracts between a servlet class and the runtime environment provided for servlet container.
  - The second describe and define the contracts between a servlet class running under the HTTP protocol and the runtime environment.
- ***Servlet like applets does not have main instead it has service() method. It receive two objects:***
  - a request object and a response object. The request object tells the servlet about the request, while the response object is used to return a response.
- ***special methods doPost or doGet are invoked for http implementations***

Request and Response





Hello World :S

Overrides

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloWorld extends HttpServlet
{
        public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
        {
                res.setContentType("text/html");
                PrintWriter out = res.getWriter();
                out.println("<HTML>");
                out.println("<HEAD><TITLE>Hello World</TITLE></HEAD>");
                out.println("<BODY>");
                out.println("<BIG>Hello World</BIG>");
                out.println("</BODY></HTML>");
        }
}
```


HelloWorld

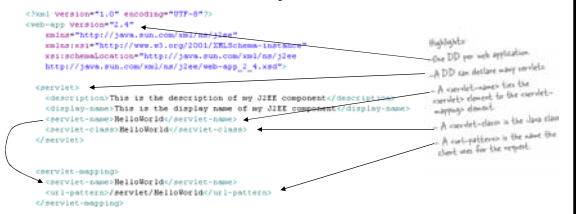
- ***You should compile your class***
  - **For J2EE**
    - javac -cp C:\Sun\SDK\lib\javaee.jar HelloWorld.java
- ***Configure the web server → tomcat***
  - **C:\apache-tomcat-5.5.20\webapps**

Hello

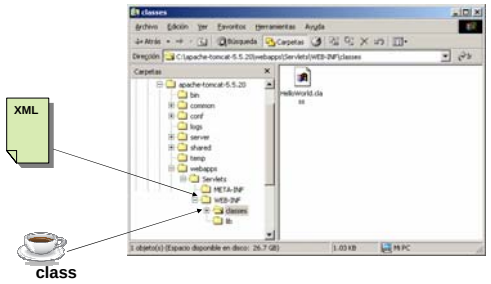
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Hello extends HttpServlet
{
        public void doGet(HttpServletRequest req, HttpServletResponse res)
                throws ServletException, IOException
        {
                res.setContentType("text/html");
                PrintWriter out = res.getWriter();
                String name = req.getParameter("name");
                out.println("<HTML>");
                out.println("<HEAD><TITLE>Hello, " + name +
"</TITLE></HEAD>");
                out.println("<BODY>");
                out.println("Hello, " + name);
                out.println("</BODY></HTML>");
        }
}
```

Deployment Descriptor

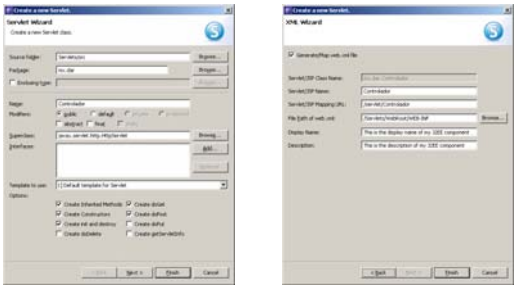
- *Our servlet would require DD which tells to server how to run it*
- *The name is web.xml putted in WEB-INF*



Directory Structure



Using an IDE

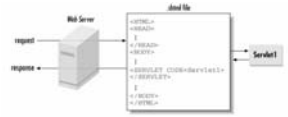


SSI

- *Also the Servlets can be embedded inside HTML through Server-Side Include (SSI)*
- *This is done by special tags:*

```
<SERVLET CODE=ServletName CODEBASE=http://server:port/dir
        initParam1=initValue1 initParam2=initValue2>
  <PARAM NAME=param1 VALUE=value1>
  <PARAM NAME=param2 VALUE=value2>
     If you see this text, it means that the web server
     providing this page does not support the SERVLET tag.
</SERVLET>
```

- *CODE = invoked name, CODEBASE = remote server*
- *PARAM let's pass any number of parameters*



Source HTML

```
<html>
<head>
    <title>Times!</title>
</head>
<body>
    The current time here is:
    <servlet code="CurrentTime"> </servlet>
    <p>
        The current time in London is:
    <servlet code="CurrentTime"> <PARAM NAME=zone VALUE=GMT>
</servlet>
    <p>
        And the current time in New York is:
    <servlet code="CurrentTime"> <PARAM NAME=zone VALUE=EST>
</servlet>
    <p>
</body>
</HTML>
```


Source CurrentTime

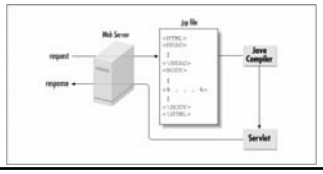
```
import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class CurrentTime extends HttpServlet
{
        public void doGet(HttpServletRequest req, HttpServletResponse res)
                                throws ServletException, IOException
        {
                PrintWriter out = res.getWriter();
                Date date = new Date();
                DateFormat df = DateFormat.getInstance();
                String zone = req.getParameter("zone");
                if (zone != null)
                {
                        TimeZone tz = TimeZone.getTimeZone(zone);
                        df.setTimeZone(tz);
                }
                out.println(df.format(date));
        }
}
```

JSP

19

JSP

- *The Java Server Pages (JSP) is a technology similar to ASP and it seems to work just like SSI, but with JSP can be embedded servlet code in the HTML page.*
- *But, jsp are treated like servlet, these are compiled, loaded and executed.*



20

Scriptlets

- *The code immersed in html receives the name of scriptlet.*
- *As you see it use special tags.*
- *All jsp directives are between <% …%>*
- *You can write expressions which are evaluated resulting in an string <%= %>*
- *Import code*
  - **<%@ import = "java.io.*,java.util.Hashtable" %>**
- *The method*
  - **<%@ method = "doPost" %>**
- *The language*
  - **<%@ language = "java" %>**

21

Hello.jsp

```
<html>
<head>
    <title>Hello</title>
</head>
<body>
    <h1>
        <%
if (request.getParameter("name") == null) {
   out.println("Hello World");
}
else {
  out.println("Hello, " + request.getParameter("name"));
}
        %>
    </h1>
</body>
</html>
```

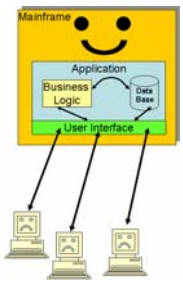
22

Web Tiers

23

One tier

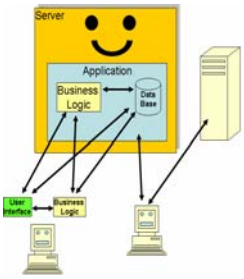
- *At the beginning there was the day of the mainframe, in this days all the applications were monolithic. This mean that the user interface, business logic and data access were all inside one central computer.*



24

Two tier

- *Later with the advent of the PC a new paradigm the client/server arises. In this approach many of the process that was done at the mainframe could be load in the PC, typically as a redistributable exe file.*
- *In this point of the original client/server approach receives the name of two-tier client/server.*



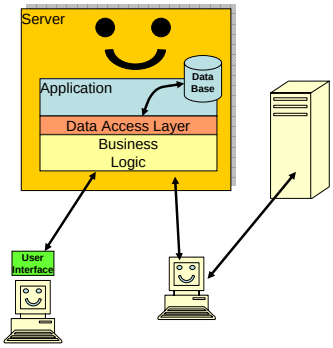
25

Three tier (1)

- *One of the biggest problem with the two-tier approach was the fragility of the application because all the business rule and data access were at the server in that way any change in the data model or business logic often requires a new deployment at the client.*
- *For this reason the approach of multi-tier is currently followed. Conceptually, an application can have any number of tiers but the most popular is the three-tier approach.*

26

Three tier (2)



27

A multi-tier approach

- *As was said an application can have any number of tier or layers, each tier provides isolation between layers. Because there are many tiers these can be distributed in different servers providing a more flexible and scalable application.*

28

A multi-tier approach

- *The evolution of a multi-tier architecture is a Distributed System where instead of layers the distributed system model exposes all functionality of the application as objects.*
- *All this is supported by encouraging the definition of component interfaces.*
- *The interface as we saw exposes which services are available. While the interface remains constant the implementation can change dramatically without affecting other components.*

29

Home work

- *Give a talk about PHP*
- *Give a talk about POJO*
- *Give a talk about JSF*
- *Give a talk about Hibernate*
- *Give a talk about AJAX*

30

Reference

- **Java Server Programming, O'Reilly**
- ***Head First Servlet and JSP, O'Reilly***

31