

Arquitectura e Integración de Aplicaciones Empresariales

Tercera Sesión Servlets

Universidad Autónoma Metropolitana
Casa abierta al tiempo  Azcapotzalco

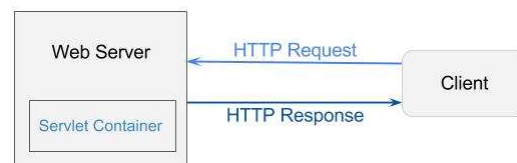
Dra. Maricela Bravo
Cubículo H-287-B
mari_clau_18@hotmail.com

¿Qué es un Servlet?

- ▶ Los Servlets son clases en Java que se ejecutan en un servidor Web, para extender las capacidades de respuesta a los clientes al utilizar las potencialidades de Java.
- ▶ Los Servlets pueden ser incluidos en servidores que soportan la API de Servlet. La API no realiza suposiciones sobre el entorno que se utiliza, como tipo de servidor o plataforma, ni del protocolo a utilizar, aunque existe una API especial para HTTP.

Servlets...

- ▶ La tarea de un servlet:
 - ▶ Leer datos explícitos enviados por el cliente a través de formas (html)
 - ▶ Leer datos implícitos enviados por el cliente a través de la cabecera del *Request*
 - ▶ Enviar datos explícitos de regreso al cliente
 - ▶ Enviar los datos implícitos al cliente a través de la cabecera del *Response*



Propiedades de los Servlets

- ▶ **Manejo de Sesiones:** Se puede hacer seguimiento de usuarios a través de distintos servlets.
- ▶ **Utilización de Cookies:** Las cookies son datos en texto plano que pueden ser guardados en el cliente.
- ▶ **Multi-hilos:** Los servlets soportan el acceso concurrente de los clientes, aunque hay que tener especial cuidado con las variables compartidas a menos que se utilice la interfaz `SingleThreadModel`.
- ▶ **Programación en Java:** Se cuenta con las herramientas y tecnologías de acceso a APIs como JDBC, RMI, etc.

Servidores que soportan la ejecución de Servlets y JSPs

Servidor	Descripción	Link
Apache Tomcat	Es un buen servidor bastante popular para páginas JSP	http://jakarta.apache.org/
JSWDK	Es un servidor sencillo y básico para probar aplicaciones JSP	http://pdhut.freesevers.com/dev_env/servletjsp/jswdk/index.htm
Sun Java System Web Server	Es un servidor web de Sun Microsystems, pensado para ejecutar JSP y Servlets, pero también con soporte para ASP, PHP o CGI	https://www.oracle.com/sun/
Macromedia JRun	Es un servidor de Java desarrollado por Macromedia (ahora Adobe). Es comercial, aunque se puede obtener una versión de prueba	http://www.adobe.com/products/jrun/
ServletExec	Aplicación comercial que sirve como módulo para ejecutar páginas JSP en servidores web populares, como Apache o IIS	http://www.newatlanta.com/products/servletexec

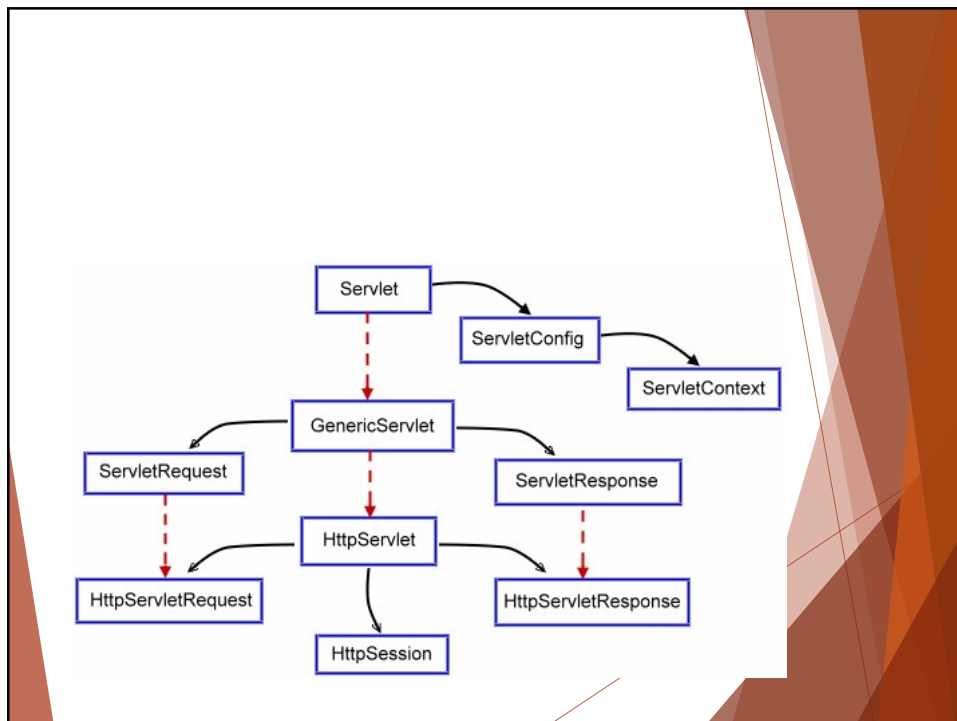
Servlets

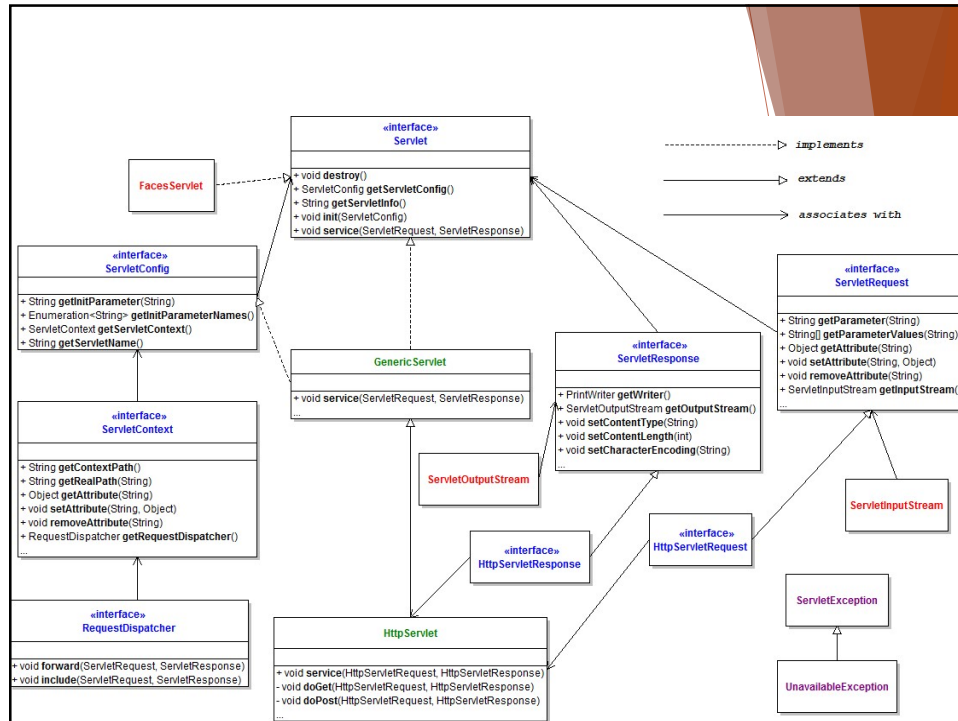
- ▶ La clase *HttpServlet* es una clase que implementa la interfaz *Servlet*, incorporando además métodos específicos para servidores Web.
- ▶ El método `service()` de la clase *HttpServlet* lanza diferentes peticiones a distintos métodos.
- ▶ Reconoce los métodos estándar en formato HTTP/1.1
- ▶ Los métodos de petición que reconoce son: GET, HEAD, PUT, POST, DELETE, OPTIONS y TRACE.
- ▶ Para cada uno Java lanza un método de tipo `doXXX()`. Por ejemplo, para GET lanza `doGet()`.

La clase HttpServlet

Los datos de la petición se pasan a todos los métodos como primer argumento de tipo HttpServletRequest

Las respuestas que pueden crear los distintos métodos se devuelven en el segundo argumento de tipo HttpServletResponse





Ejemplo de Servlet

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(name = "TestServlet", urlPatterns = {"/TestServlet"})
public class TestServlet extends HttpServlet
{
```

```

protected void processRequest (HttpServletRequest request,
HttpServletRequest response)
    throws ServletException, IOException
    {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Prueba de Servlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Contexto del servlet " + request.getContextPath()
+ "</h1>");
        out.println("</body>");
        out.println("</html>");
    } finally {
        out.close();
    }
    }

```

```

@Override
protected void doGet (HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
    }

@Override
protected void doPost (HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
    }

@Override
public String getServletInfo()
    {
    return "Short description";
    } // </editor-fold>
}

```

Lectura de datos desde el formulario

```
<html>
  <head>
    <title></title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    <div>
      <form action="tresParams">
        Primer parametro: <input type="text" name="param1"> <br>
        Segundo parametro: <input type="text" name="param2"> <br>
        Tercer parametro: <input type="text" name="param3"> <br>
        <center> <input type="submit"></center>
      </form>
    </div>
  </body>
</html>
```

```
@WebServlet(name = "tresParams", urlPatterns = {"/tresParams"})
public class tresParams extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws
    ServletException, IOException
    {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            String title = "Lectura de los parametros del Request";
            String docType = "<!DOCTYPE HTML PUBLIC \"/>";
            out.println(docType +
                "<HTML>\n" +
                "<HEAD><TITLE>" + title + "</TITLE></HEAD>\n" +
                "<BODY BGCOLOR=\"#FDF5E6\">\n" +
                "<H1 ALIGN=\"CENTER\">" + title + "</H1>\n" +
                "<UL>\n" +
                "  <LI><B>param1</B>: "
                + request.getParameter("param1") + "\n" +
                "  <LI><B>param2</B>: "
                + request.getParameter("param2") + "\n" +
                "  <LI><B>param3</B>: "
                + request.getParameter("param3") + "\n" +
                "</UL>\n" +
                "</BODY></HTML>");
        } finally {
            out.close(); }
    }
}
```

Ejercicio

- ▶ Desarrollar una página HTML que lea el nombre y fecha de cumpleaños de una persona y que mediante un servlet genere como respuesta el nombre de la persona y la edad que tiene.