
INICIALIZACIÓN DE SERVLET COLABORACIÓN ENTRE SERVLETS

Eduard Lara

INDICE

1. Parámetros iniciales del servlet
2. Recuperar los parámetros en el servlet
3. Parámetros iniciales de contexto
4. Lanzamiento de solicitudes
5. Método Forward
6. Método Include

1. PARÁMETROS INICIALES DEL SERVLET

- Un servlet puede tener cualquier número de parámetros de inicialización.
- El método init debe obtener los valores a partir del objeto ServletConfig.
- El contenedor web crea el objeto de configuración basándose en los parámetros de inicialización especificados en el descriptor de despliegue
- En el archivo web.xml debemos declarar los parámetros, tal como muestra la imagen.

```
<init-param>  
    <param-name>Oferta</param-name>  
    <param-value>Hoy todos los monitores al 10% de dto.</param-value>  
</init-param>
```

1.1. API DE SERVLETCONFIG

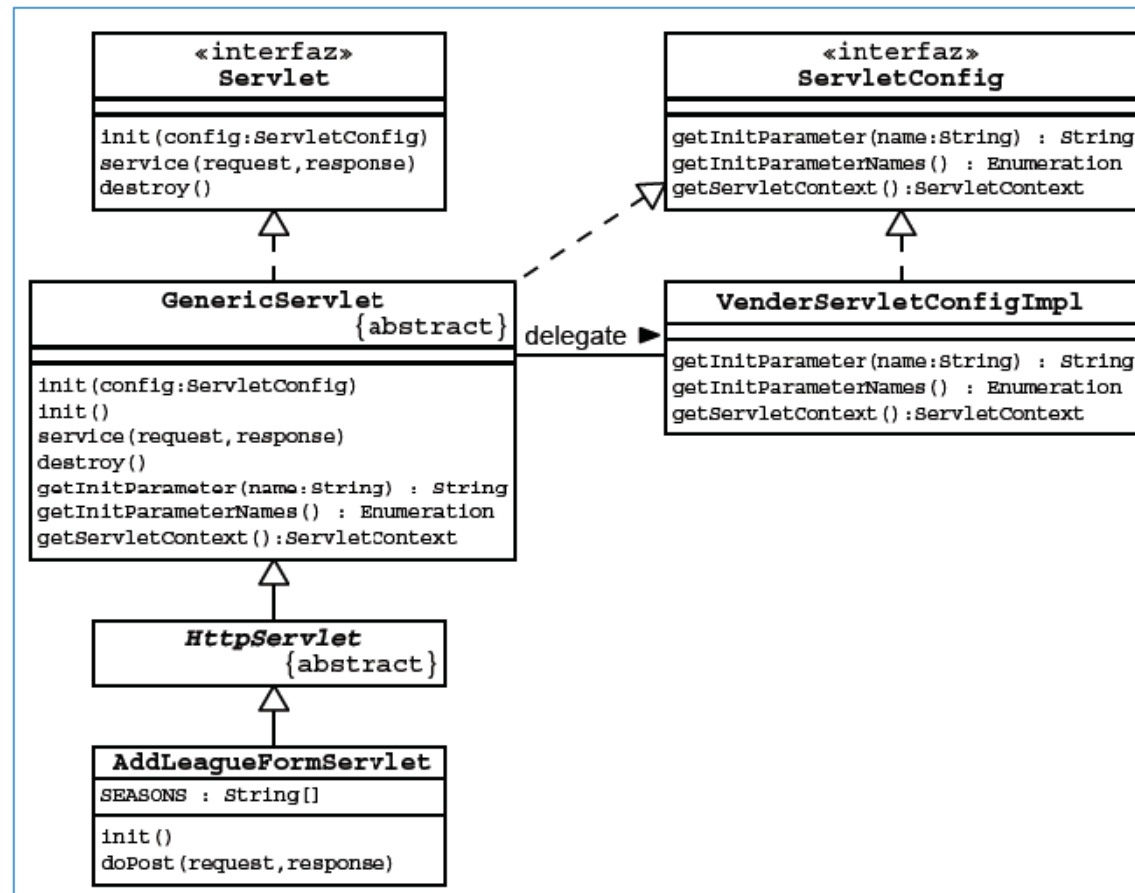
- Cada contenedor web debe implementar la interfaz ServletConfig.
- Las instancias de esta clase se pasan al método init(ServletConfig) definido en la interfaz de Servlet.
- La clase GenericServlet (suministrada en la API de servlet) implementa la interfaz de Servlet.
- Implementa el método init(ServletConfig), que almacena el objeto de configuración (delegate) y después llama al método init().
- Se trata de un método init sin argumentos que se sustituye en las clases de servlet.

1.1. API DE SERVLETCONFIG

- La interfaz `ServletConfig` proporciona el método `getInitParameter`, que permite recuperar los parámetros de inicialización del servlet.
- Por motivos prácticos, la clase `GenericServlet` también implementa la interfaz `ServletConfig`, y esos métodos delegan las llamadas en el objeto de configuración almacenado.
- Ello simplifica el código de servlet, porque permite llamar directamente al método `getInitParameter` (sin acceso directo al objeto de configuración).

1.1. API DE SERVLETCONFIG

La siguiente imagen ilustra la API de servlet y su relación con la interfaz ServletConfig.



2. RECUPERAR LOS PARÁMETROS EN EL SERVLET

Para recuperar los parámetros iniciales utilizamos el método `init(ServletConfig sc)` del servlet tal y como muestra la imagen:

```
public void init(ServletConfig config) throws ServletException {  
    // recuperamos el parametro inicial del servlet del objeto ServletConfig  
    String mensaje = config.getInitParameter("Oferta");  
    // Recuperar el contexto del servlet es lo mismo que el contexto de la aplicacion  
    ServletContext aplicacion = config.getServletContext();  
    aplicacion.setAttribute("msg", mensaje);  
    System.out.println(">>>>>>>>>>>>>>> Se ha creado la instancia del servlet");  
}
```

2. RECUPERAR LOS PARÁMETROS EN EL SERVLET



RECUERDA QUE...

- Los parámetros iniciales del servlet son datos que se recogen en el momento de la inicialización de este.
- Se declaran en el descriptor de despliegue web.xml y se recuperan en el método init().

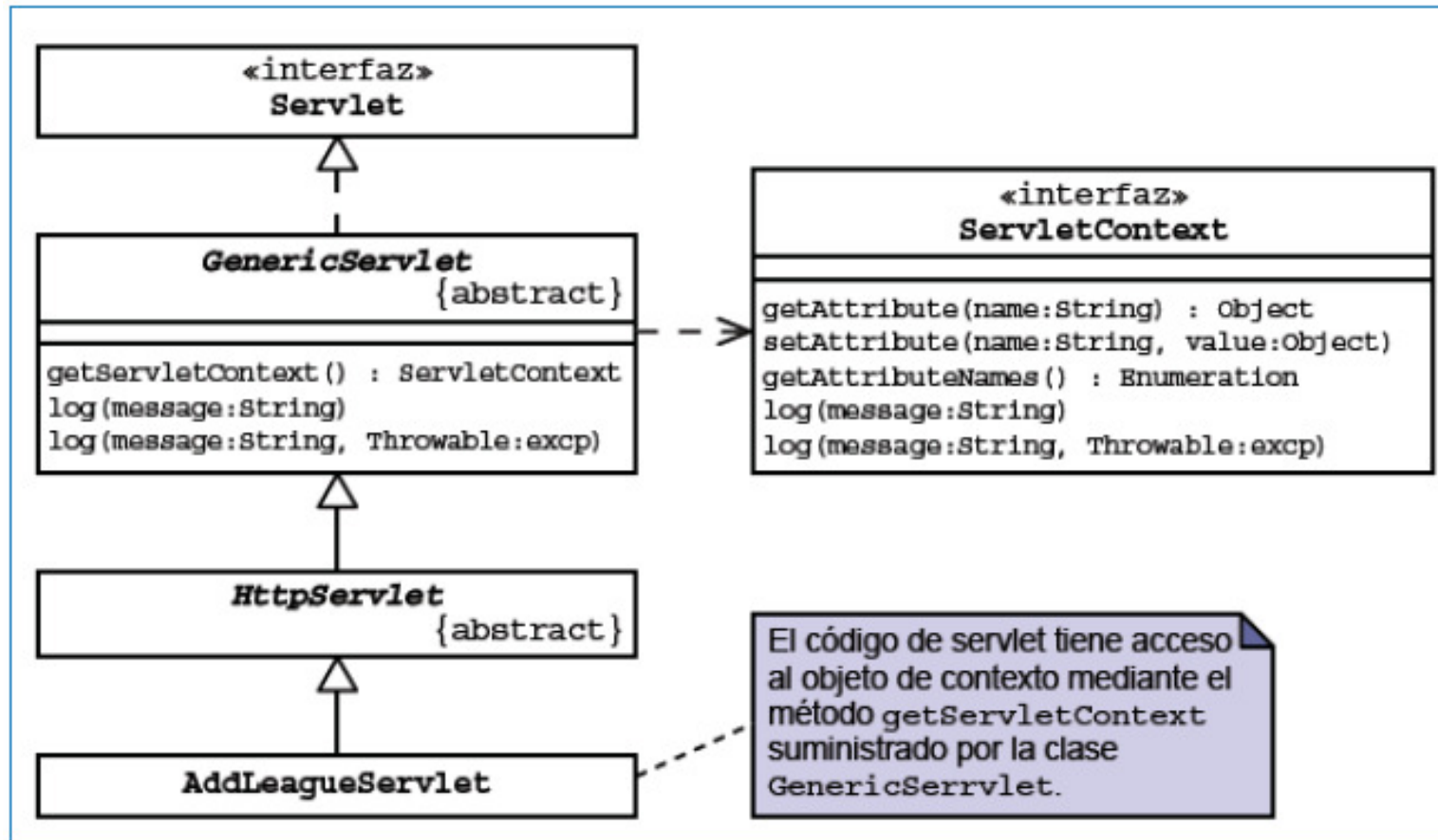
3. PARÁMETROS INICIALES DE CONTEXTO

- Una aplicación web es una colección autocontenida de recursos estáticos y dinámicos: páginas HTML, archivos multimedia, archivos de datos y de recursos, servlets (y páginas de JSP) y otros objetos y clases Java auxiliares.
- El descriptor de despliegue de la aplicación web sirve para especificar la estructura y los servicios utilizados por una aplicación web.
- Un objeto `ServletContext` es la representación de tiempo de ejecución de la aplicación web.
- El objeto de contexto es accesible a todos los servlets en una aplicación web.

3. PARÁMETROS INICIALES DE CONTEXTO

- En el objeto de contexto se pueden almacenar atributos de ámbito de contexto mediante el método `setAttribute`. Los atributos pueden recuperarse del objeto de contexto mediante el método `getAttribute`.
- También es posible eliminar un atributo de contexto con el método `removeAttribute`. El ámbito de contexto también se denomina ámbito de aplicación.
- El método `getServletContext` lo suministra la clase `GenericServlet`, ampliada por la clase `HttpServlet` y las clases de servlet del usuario.

3. PARÁMETROS INICIALES DE CONTEXTO



3. PARÁMETROS INICIALES DE CONTEXTO

En la siguiente figura, se puede ver cómo se hace la declaración de parámetros de contexto en el web.xml:

```
<context-param>  
  <param-name>OfertaMes</param-name>  
  <param-value>Este mes las impresoras con un 25% de dto</param-value>  
</context-param>
```

4. LANZAMIENTO DE SOLICITUDES

- El lanzamiento de solicitudes permite a un servlet lanzar una solicitud a otro servlet, página JSP o página HTML, que será la responsable de cualquier procesamiento posterior y de generar la respuesta.
- El API Java Servlet tiene una interfaz especial llamada `javax.servlet.RequestDispatcher` para este propósito.
- Esta interfaz tiene dos métodos que le permiten delegar el procesamiento de solicitudes-respuestas en otro recurso, después de que el servlet llamante haya finalizado cualquier procesamiento preliminar:
 - El método `forward()`
 - El método `include()`.

5. MÉTODO FORWARD

- El método **forward()**; permite reenviar la solicitud a otro servlet o página JSP, o a un archivo HTML en el servidor; este recurso acepta la responsabilidad de producir la respuesta.

**public void forward(ServletRequest request,
ServletResponse response) throws
ServletException, java.io.IOException**

Difiere de `HttpServletResponse.sendRedirect()`, que redirige la petición con la ayuda del navegador.

6. MÉTODO INCLUDE

- El método `include()`; permite incluir el contenido producido por otro recurso en la respuesta del servlet llamante.

```
public void include(ServletRequest request,  
ServletResponse response) throws  
ServletException, java.io.IOException
```