LISTENERS

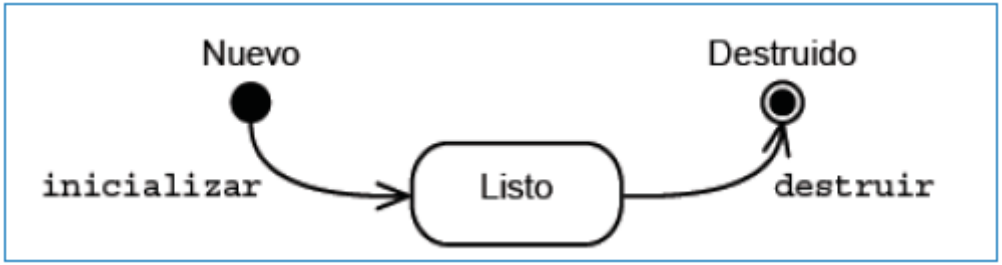
Eduard Lara

INDICE

1. Listener de contexto
2. Listener de atributos de contexto
3. Listener de sesión
4. Listener de atributos de sesión

1. LISTENER DE CONTEXTO

- La aplicación web (representada por el objeto de contexto) tiene un ciclo de vida gestionado por el contenedor web. Este ciclo de vida es similar al del servlet.

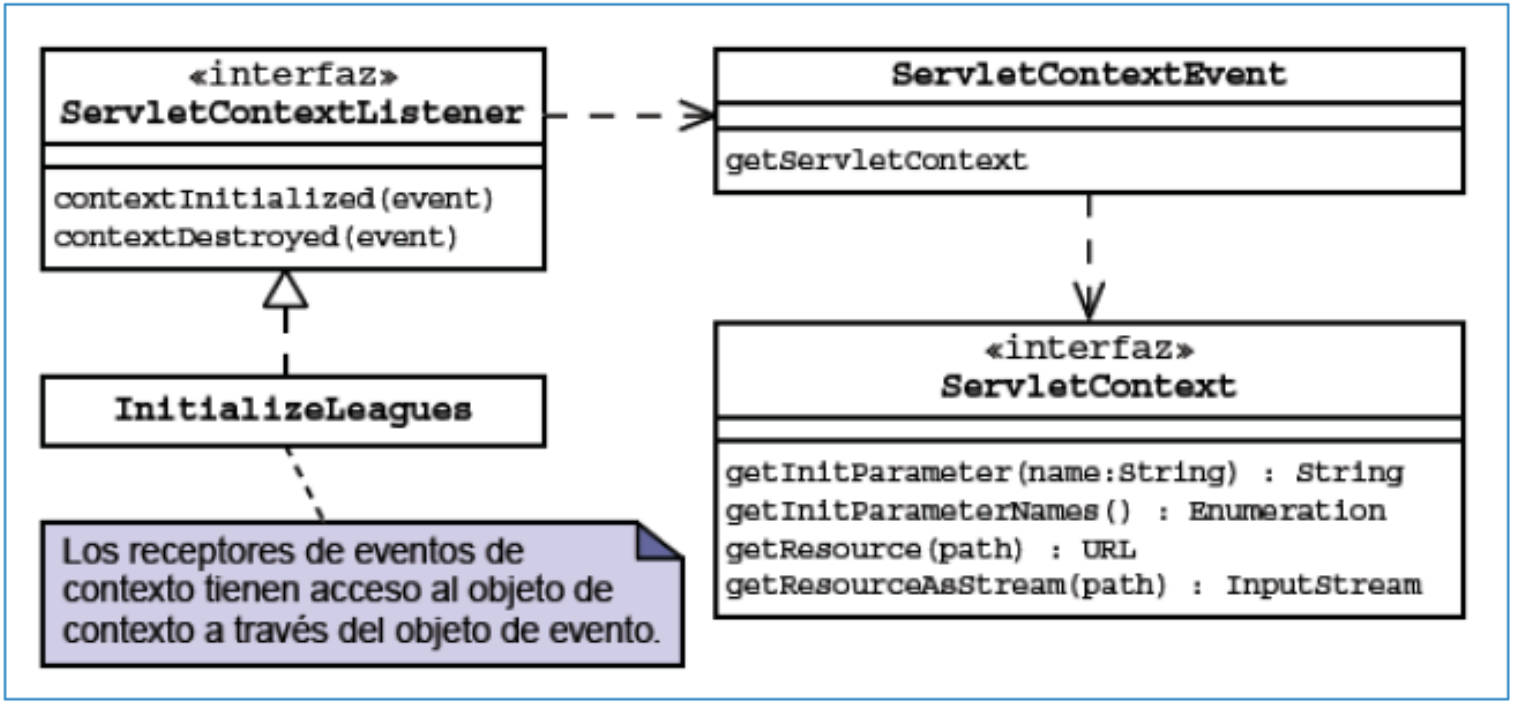


- Cuando se inicia el contenedor web, se inicializa cada aplicación web. Cuando se cierra el contenedor web, se destruye cada aplicación web.
- Para recibir estos eventos del ciclo de vida de la aplicación web se necesita un receptor de eventos de contexto de servlet.

1. LISTENER DE CONTEXTO

- Normalmente, los datos compartidos de la aplicación deben hallarse en memoria antes de ejecutar cualquier solicitud HTTP en la aplicación web.
- La interfaz ServletContextListener tiene dos métodos.
  - El contenedor web invoca el método contextInitialized cuando se ha iniciado la aplicación web.
  - El contenedor web invoca el método contextDestroyed cuando se está cerrando la aplicación web.
- Ambos métodos incluyen un argumento de evento. El objeto de evento proporciona acceso al objeto de contexto mediante el método getServletContext. Los receptores de eventos de contexto deben implementar esta interfaz

1. LISTENER DE CONTEXTO



1. LISTENER DE CONTEXTO

Configuración del receptor de eventos

Debemos configurar el listener de contexto en el descriptor de despliegue web.xml.

```
<listener>
    <description>ServletContextListener</description>
    <listener-class>app.web.ListenerAplicacion</listener-class>
</listener>
```

Recuperar los parámetros de contexto

Los parámetros de contexto los recuperamos en el listener de la aplicación. Concretamente en el método contextInitialized.

```
public void contextInitialized(ServletContextEvent sce) {
    ServletContext aplicacion = sce.getServletContext();
    String mes = aplicacion.getInitParameter("OfertaMes");
    aplicacion.setAttribute("mes", mes);
}
```

1. LISTENER DE CONTEXTO

```
public class MiServletContextListener implements ServletContextListener{

    @Override
    public void contextInitialized(ServletContextEvent sce) {
        throw new UnsupportedOperationException("Not supported yet.");
    }

    @Override
    public void contextDestroyed(ServletContextEvent sce) {
        throw new UnsupportedOperationException("Not supported yet.");
    }
}
```

```
    <listener>
        <listener-class>listener.MiServletContextListener</listener-class>
    </listener>
</web-app>
```

RECUERDA QUE...

- Los parámetros de contexto se consideran parámetros iniciales de la aplicación.
- Se declaran en el web.xml como <context-param>
- Se recuperan a través de un Listener de contexto.

2. LISTENER DE ATRIBUTOS DE CONTEXTO

```
public class MiServletContextAttributeListener implements ServletContextAttributeListener{

    @Override
    public void attributeAdded(ServletContextAttributeEvent scae) {
        throw new UnsupportedOperationException("Not supported yet.");
    }

    @Override
    public void attributeRemoved(ServletContextAttributeEvent scae) {
        throw new UnsupportedOperationException("Not supported yet.");
    }

    @Override
    public void attributeReplaced(ServletContextAttributeEvent scae) {
        throw new UnsupportedOperationException("Not supported yet.");
    }
}
```

```
    <listener>
        <listener-class>listener.MiServletContextAttributeListener</listener-class>
    </listener>
</web-app>
```

2. LISTENER DE ATRIBUTOS DE CONTEXTO

- Si un atributo de contexto es agregado, se ejecutará el listener attributeAdded().
- Si un atributo de contexto es actualizado, se ejecutará el listener attributeReplaced().
- Si un atributo de contexto es eliminado, se ejecutará el listener attributeRemoved().

```
request.setAttribute("url", "espai.es"); //Es ejecutado attributeAdded()
request.setAttribute("url", "espai.es"); //Es ejecutado attributeReplaced()
request.removeAttribute("url");          //Es ejecutado attributeRemoved()
```

3. LISTENER DE SESION

```
public class MiHttpSessionListener implements HttpSessionListener{

    @Override
    public void sessionCreated(HttpSessionEvent hse) {
        throw new UnsupportedOperationException("Not supported yet.");
    }

    @Override
    public void sessionDestroyed(HttpSessionEvent hse) {
        throw new UnsupportedOperationException("Not supported yet.");
    }
}
```

```
    <listener>
        <listener-class>listener.MiHttpSessionListener</listener-class>
    </listener>
</web-app>
```

3. LISTENER DE SESION

- Si una nueva sesion es creada (con request.getSession()) se ejecutará el listener sessionCreated()
- Si una session es destruida (por timeout de session o session.invalidate()), el listener sessionDestroyed() será ejecutado.

```
HttpSession session = request.getSession(); //Se ejecutará sessionCreated()
session.setAttribute("url", "espai.es");
session.invalidate();                       //Se ejecutará sessionDestroyed()
```

4. LISTENER DE ATRIBUTOS DE SESION

```
public class MiHttpSesionAttributeListener implements HttpSessionAttributeListener {

    @Override
    public void attributeAdded(HttpSessionBindingEvent hsbe) {
        throw new UnsupportedOperationException("Not supported yet.");
    }

    @Override
    public void attributeRemoved(HttpSessionBindingEvent hsbe) {
        throw new UnsupportedOperationException("Not supported yet.");
    }

    @Override
    public void attributeReplaced(HttpSessionBindingEvent hsbe) {
        throw new UnsupportedOperationException("Not supported yet.");
    }
}
```

```
    <listener>
        <listener-class>listener.MiHttpSessionListener</listener-class>
    </listener>
</web-app>
```

4. LISTENER DE ATRIBUTOS DE SESION

- Si un atributo de sesion es agregado, se ejecutará el listener attributeAdded().
- Si un atributo de sesión es actualizado, se ejecutará el listener attributeReplaced().
- Si un atributo de sesión es eliminado, se ejecutará el listener attributeRemoved().

```
HttpSession session = request.getSession();
session.setAttribute("url", "espai.es"); //Se ejecutará attributeAdded()
session.setAttribute("url", "espai.es");//Se ejecutará attributeReplaced()
session.removeAttribute("url"); //Se ejecutará attributeRemoved()
```