Los problemas con HttpSession invalidate son muy comunes cuando trabajamos con Java web. Todo el mundo se maneja con el concepto de session , pero siempre quedan algunas dudas a la hora de manejarlo a detalle. Una de las situaciones problemáticas más habituales es que hacer cuando una session caduca. Vamos a crear un ejemplo sencillo de manejo de session con un asistente.

HttpSession Invalidate asistente

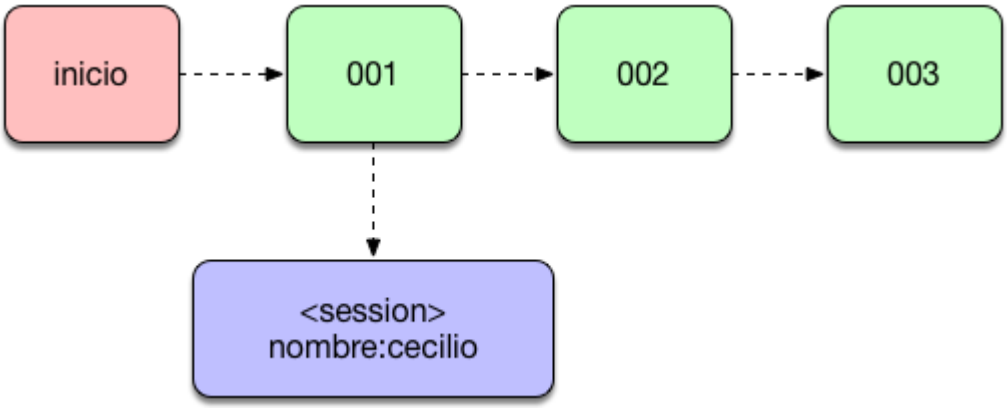
Nuestro asistente se compone de 4 paginas

- inicio.jsp
- 001.jsp
- 002.jsp
- 003.jsp

El asistente comienza con la página de inicio que invoca a la página 001.jsp . Esta página se encarga de almacenar un nombre en la session.



Una vez almacenado en la sesión nos permite navegar a la página 002.jsp y comprobar que se muestra correctamente.El último paso es navegar a la página 003.jsp que vuelve e mostrar la misma información.



Vamos a ver su código:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
empezamos
<a href="asistente/001.jsp">asistente</a>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>

<% session.setAttribute("nombre", "cecilio"); %>
nombre salvado
<a href="002.jsp">mostrar nombre</a>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>

<%=session.getAttribute("nombre")%>
```

```
<a href="003.jsp">mostrar nombre</a>
</body>
</html></pre>

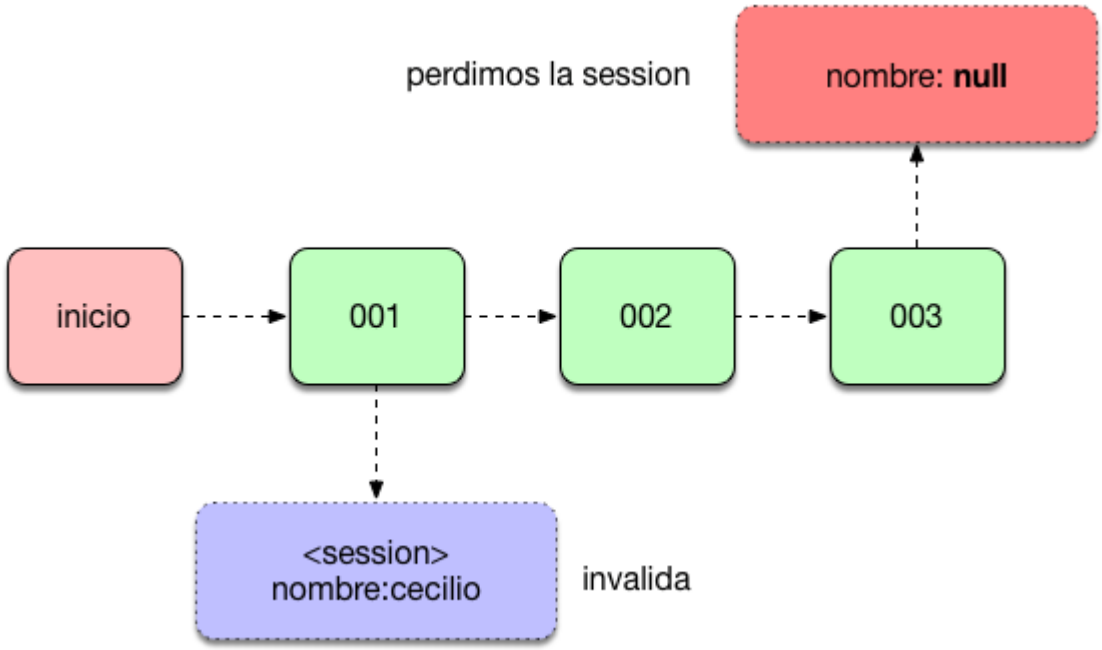

<pre>



<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>

<%=session.getAttribute("nombre")%>
</body>
```

El código es muy sencillo y funcionará correctamente. El problema viene cuando estamos en un paso intermedio del asistente y dejamos que pase mucho tiempo y nuestra session caduca.



En este caso cuando el asistente intente avanzar se producirá un error.



Para poder ver de forma cómoda el error vamos a modificar el tiempo de timeout de nuestro web.xml a 1 minuto.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" version="3.1">
```

```
<display-name>JavaSessionInvalidate</display-name>
<welcome-file-list>
<welcome-file>index.html</welcome-file>
<welcome-file>index.htm</welcome-file>
<welcome-file>index.jsp</welcome-file>
<welcome-file>default.html</welcome-file>
<welcome-file>default.htm</welcome-file>
<welcome-file>default.jsp</welcome-file>
</welcome-file-list>
<session-config>
<session-timeout>1</session-timeout>
</session-config>
</web-app>
```

Tenemos un problema importante ya que no podemos presentar ese tipo de información al usuario.

HttpSession invalidate y Servlet Filters

Para solventarlo , podemos diseñar un filtro que se encargue de proteger todas las URLS del asistente salvo la de inicio. Este filtro comprobará si tenemos una session activa y en el caso de que no este disponible nos redireccionará a la página de inicio.

```
package com.arquitecturajava;

import java.io.IOException;
import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
```

```
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.annotation.WebFilter;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
* Servlet Filter implementation class SessionFilter
*/
@WebFilter("/asistente/*")
public class SessionFilter implements Filter {
public void doFilter(ServletRequest request, ServletResponse response,
FilterChain chain) throws IOException, ServletException {

HttpServletRequest peticion=(HttpServletRequest)request;
HttpSession session = peticion.getSession(false);

HttpServletResponse respuesta=(HttpServletResponse)response;



if(session != null) {
chain.doFilter(request, response);
} else {
respuesta.sendRedirect("../inicio.jsp");
}

}
```

```
@Override
public void destroy() {
// TODO Auto-generated method stub

}

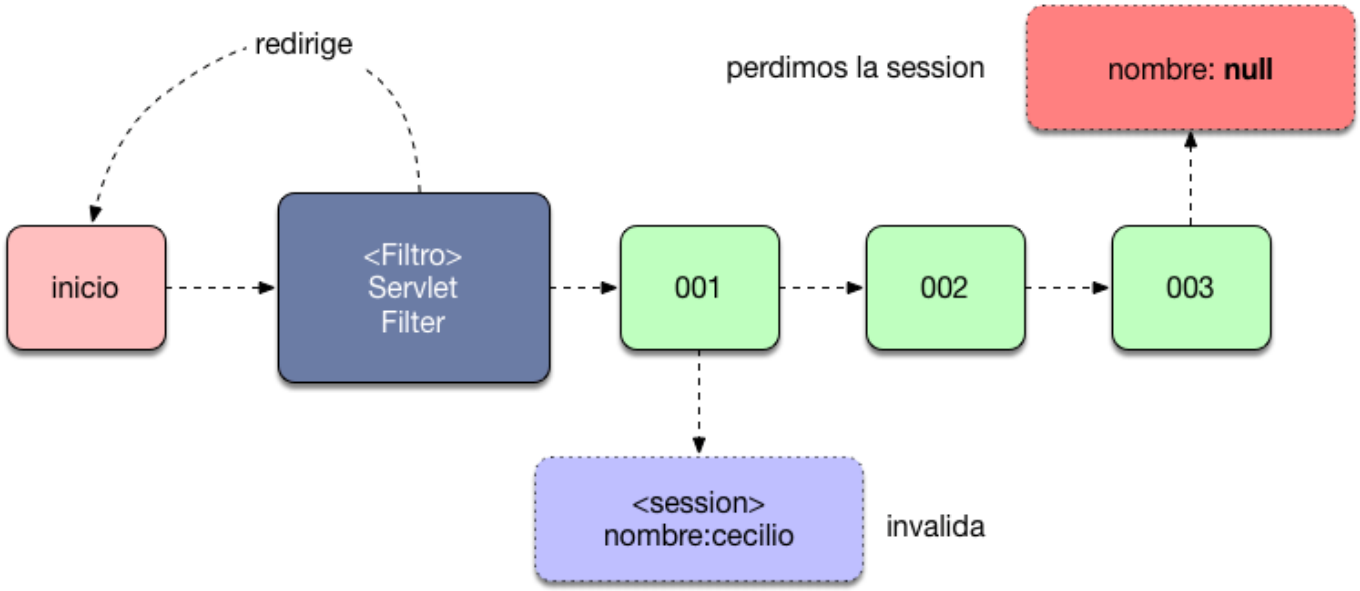
@Override
public void init(FilterConfig arg0) throws ServletException {
// TODO Auto-generated method stub

}
}
```

De esta manera conseguiremos que de forma automática si la session caduca nos redireccione a la página de inicio.



La nueva estructura de páginas y servlets será:



Acabamos de añadir un filtro que se encarga de gestionar la invalidación de sesiones. Cada framework Java EE Web tiene su solución concreta pero el concepto de filtro es muy común en la mayoría para realizar este tipo de funcionalidades.

Otros artículos relacionados

- Java HttpSession Timeout
- Usando Java Session en aplicaciones web
- Java ServletContext y aislamiento.
- Oracle HttpSession