MANEJO DE SESIONES

Eduard Lara

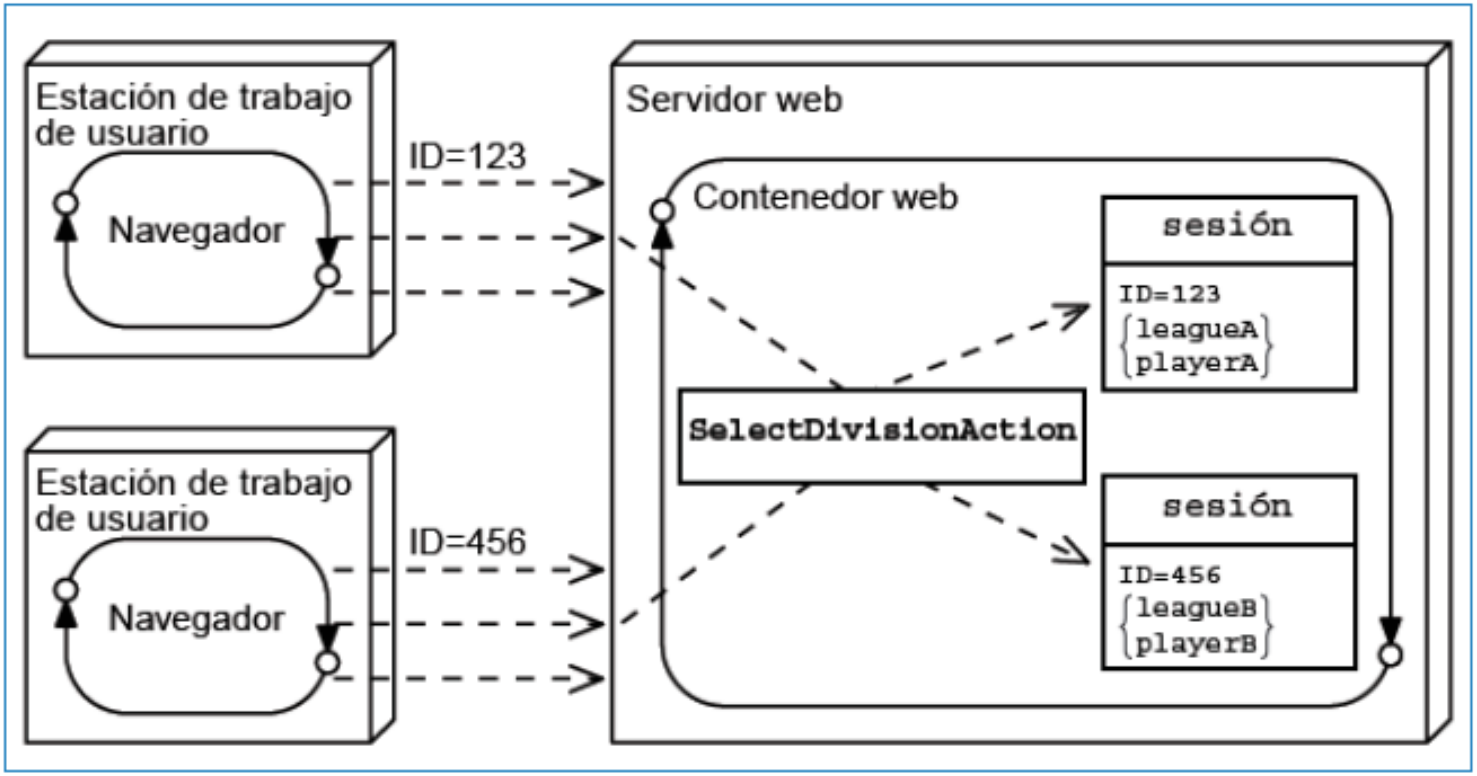
INDICE

1. Manejo de sesiones
2. API de HTTPSession
3. nstalación de Eclipse
4. Crear proyecto J2EE en Eclipse

1. MANEJO DE SESIONES

- HTTP es un protocolo sin datos de estado.
- Cada conexión de mensaje de solicitud y respuesta es independiente de todas las demás.
- Esto significativo, porque entre solicitud y solicitud (del mismo usuario) el servidor HTTP no conserva una referencia a la solicitud anterior.
- Por tanto, el contenedor web debe establecer un mecanismo para almacenar la información de sesión de un usuario determinado.
- Las sesiones constituyen un mecanismo para almacenar datos específicos del cliente a lo largo de diversas solicitudes HTTP.

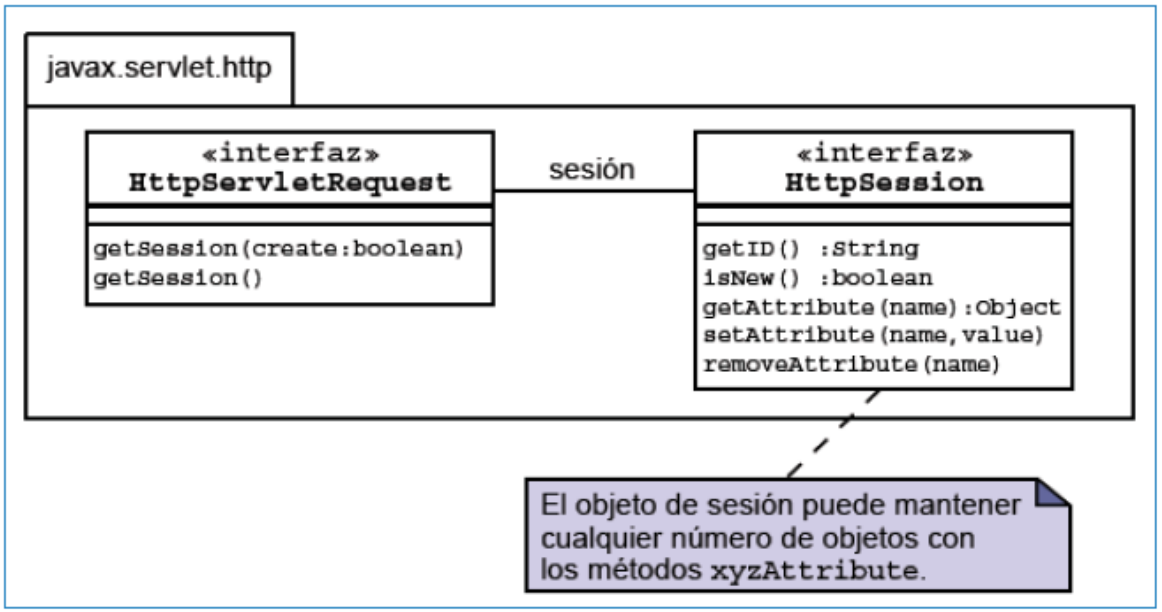
1. MANEJO DE SESIONES



Cada cliente recibe un identificador de sesión único que el contenedor web utiliza para identificar el objeto de sesión de ese usuario.

2. API DE HTTPSESSION

- La especificación de Servlet proporciona una interfaz HttpSession que permite almacenar atributos de sesión.
- Puede almacenar, recuperar y eliminar atributos del objeto de sesión. El servlet tiene acceso al objeto de sesión con el método getSession del objeto HttpServletRequest.



3. CREAR UNA SESIÓN

- La interfaz HttpServletRequest también tiene un método getSession(boolean).
- Si llama a este método con un argumento true, se crea un objeto de sesión nuevo si aún no existía.
- Si llama a este método con un argumento false, se devuelve null si aún no existía una sesión.
- Podemos considerar que llamar al método getSession() equivale a llamar a getSession(true).

```
HttpSession sesion = request.getSession();
```

4. ALMACENAMIENTO DE ATRIBUTOS DE SESIÓN

- A través del método setAttribute(nombre, valor) podemos almacenar datos (atributos) en el contexto de sesion.

```
sesion.setAttribute("carrito", miCarro);
```

- Estos atributos permanecerán todo el tiempo que dure la sessión.
- La diferencia con los atributos almacenados en la petición (request) es que estos solo existen hasta que se complete la respuesta en el cliente.

5. ACCESO A ATRIBUTOS DE SESIÓN

- Para recuperar los atributos de la sesión utilizamos el método getAttribute(nombre).
- Esto nos devuelve un Object por lo cual habrá que hacer el casting al tipo de dato adecuado.

```
Carrito miCarro = (Carrito) sesion.getAttribute("carrito");
```

6. DESTRUCCIÓN DE LA SESIÓN

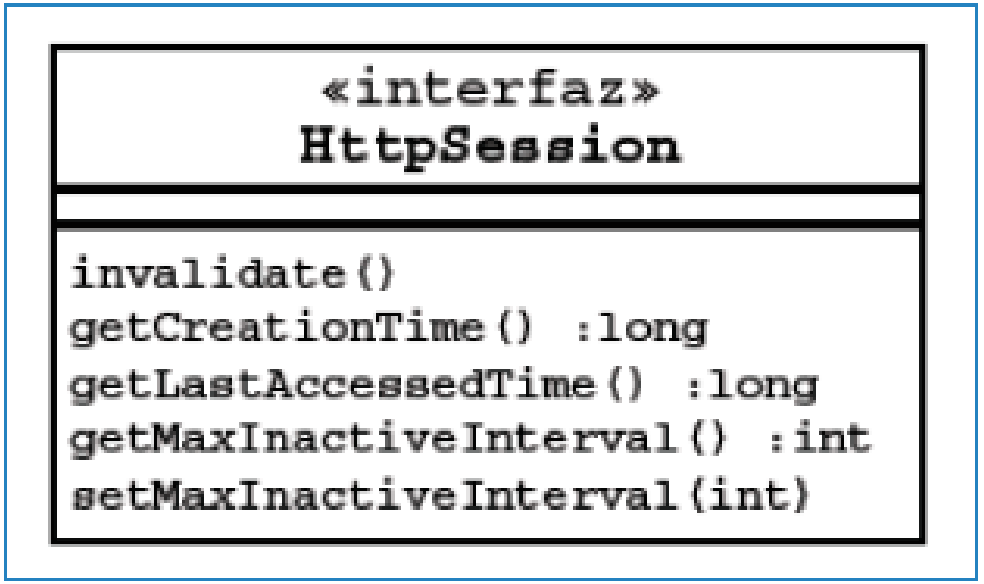
Cuando la aplicación web completa una sesión, el controlador puede destruir (efectivamente) la sesión con el método invalidate.

Hay otros dos mecanismos para destruir una sesión, ambos gestionados por el contenedor web.

1) Puede configurar un parámetro de tiempo de espera en el descriptor de despliegue. El valor del elemento session-timeout debe ser un número entero que represente el número de minutos que puede durar una sesión si el usuario la ha dejado inactiva.

6. DESTRUCCIÓN DE LA SESIÓN

2) El segundo mecanismo le permite controlar la longitud del intervalo de inactividad para un objeto de sesión específico. Puede utilizar el método setMaxInactiveInterval para cambiar el intervalo de inactividad (en segundos) del objeto de sesión.



6. DESTRUCCIÓN DE LA SESIÓN

RECUERDA QUE...

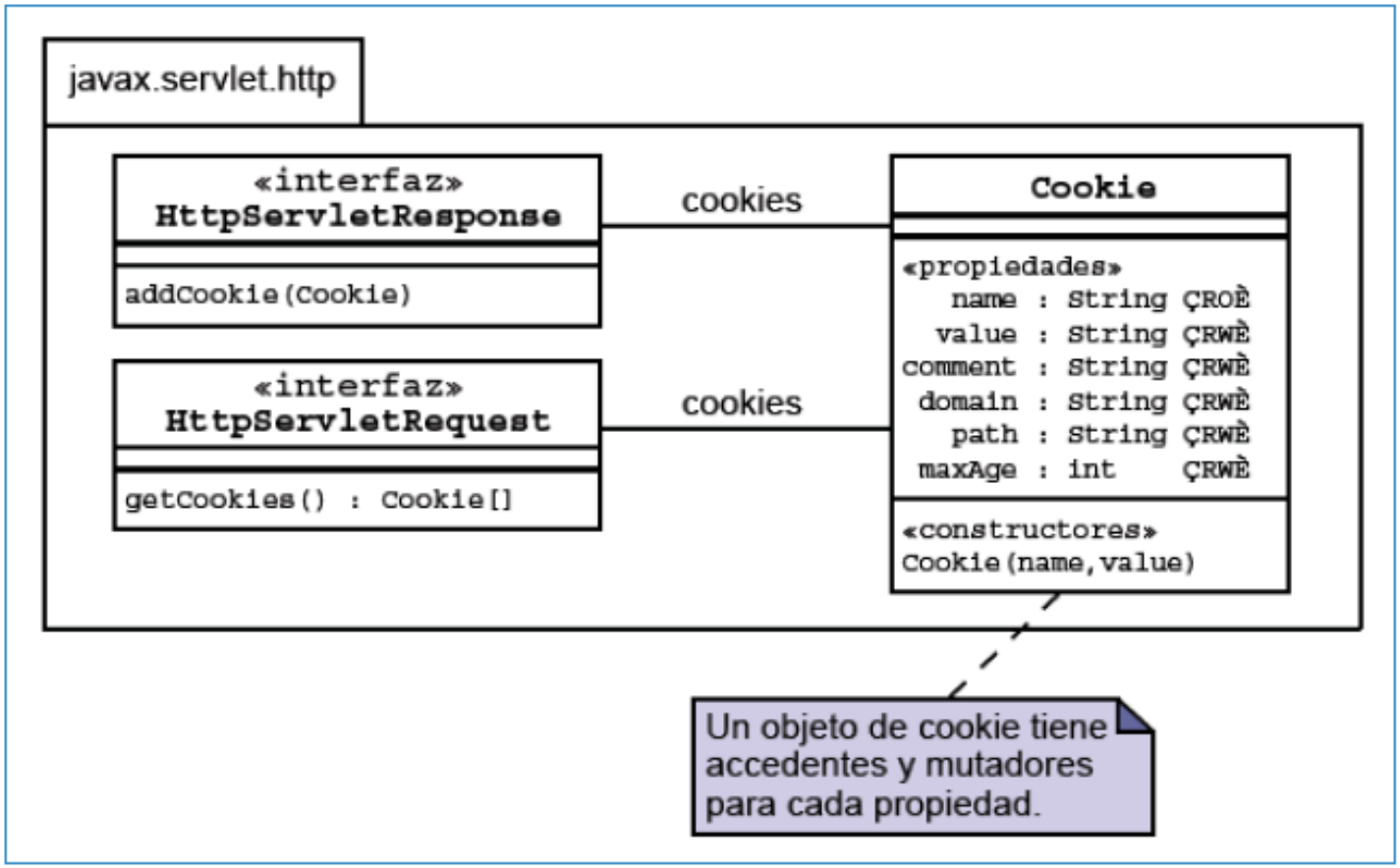
- Las sesiones permiten almacenar datos durante varias peticiones del mismo cliente.
- Se crea un objeto sesión por cada cliente que la necesita.
- Las sesiones utilizan Cookies para almacenar el número de la sesión.

7. COOKIES

- Las cookies son el mecanismo de seguimiento de sesiones más utilizado.
- Son archivos de texto almacenados en la maquina del cliente con ayuda de un navegador Web.
- Las cookies almacenan información utilizando los pares nombre/valor y la devuelven al servidor que la creó en posteriores peticiones.

7. COOKIES

API DE COOKIE



8. FUNCIONAMIENTO DE LAS COOKIES

- El servidor crea una cookie, la llena de información relevante y la envía al navegador del cliente.
- El cliente almacena esta cookie en el disco duro y éste envía la cookie de nuevo al servidor en posteriores peticiones.
- Todas las cookies tienen una fecha de caducidad establecida, por lo que son temporales, pero se puede establecer esta fecha en un fututo lejano si se necesita.

8. FUNCIONAMIENTO DE LAS COOKIES

- **Para crear una cookie** se crea un objeto de la clase Cookie:

  Cookie micookie = new Cookie("nombre", "valor");

- Las cookies tienen un periodo tras el cual expiran, **para poder establecer el tiempo de permanencia:**

  micookie.setMaxAge(int valor);

  - El valor representa los segundos transcurridos desde que se crea la cookie.
  - Ejemplos:
    - 24*60*60 establece 24 horas
    - 365*24*60*60 establece un año
    - 2*365*24*60*60 establece dos años

8. FUNCIONAMIENTO DE LAS COOKIES

- Una vez creada la cookie y habiendo establecido el tiempo de permanencia ya podemos **almacenar la cookie en el navegador del cliente**, para ello:
  - response.addCookie(micookie);

- **Para eliminar una cookie** establecemos el tiempo a cero.
  - micookie.setMaxAge(0);

8. FUNCIONAMIENTO DE LAS COOKIES

- Si queremos **saber todas las cookies que tenemos almacenadas en el navegador del cliente** crearemos un array de cookies donde las almacenaremos todas.
    - Cookie lista_cookies[] = request.getCookies();

- **Para localizar una cookie determinada** después de obtenerlas todas en un array de cookies. Lógicamente tendremos que recorrer dicho array y comparar de una en una. Tenemos dos opciones:
    - Buscar por nombre: lista_cookies[i].getName();
    - Buscar por valor: lista_cookies[i].getValue();

8. FUNCIONAMIENTO DE LAS COOKIES

- **Para modificar o actualizar el valor de una cookie**:
  - micookie.setValue("nuevo valor");

- Las cookies nos dan la opción de poner un comentario en ellas que muchas veces nos pueden servir de gran utilidad.
  - **Para establecer un comentario:**
    - micookie.setComment("comentario");
  - **Para leer un comentario:**
    - micookie.getComment();

8. FUNCIONAMIENTO DE LAS COOKIES

- Para saber que cookies son nuestras podemos **especificar nuestro dominio**.
  - micookie.setDomain();
  - Ejemplo: Si mi dominio es espai.es
    - micookie.setDomain(espai.es);

RECUERDA QUE...

- Las cookies son pequeños ficheros de texto que almacenan la información en el navegador web del usuario.
- Se utilizan para el manejo de sesiones.
- Cuando llega la petición de un cliente en ella viajan las cookies de nuestro dominio.