COOKIES

Tiempo estimado: 20m

En este punto del curso, se hace necesario presentar el concepto de *cookie*, medio a través del cual solicitar a los clientes que almacenen cierta información de la aplicación en sus discos duros con objeto de remitírsela a la aplicación cada vez que realicen una petición HTTP.

Primero, se introduce y se describe las distintas partes que tiene una *cookie*: su nombre, su valor y sus parámetros. A continuación, se presenta el componente de *middleware* cookie-parser que facilita, a la aplicación Express, el acceso a las *cookies* remitidas por los clientes. Después, se muestra cómo acceder a las *cookies* tanto en los mensajes HTTP recibidos de los clientes como en los que remite el servidor como respuesta. Finalmente, cómo configurar la administración de *cookies* en varios navegadores webs.

Al finalizar la lección, el estudiante sabrá:

- Qué es una *cookie*.
- Cuál es el objeto de una *cookie*.
- Qué tipos de *cookie* hay.
- Cuáles son los parámetros de las *cookies*.
- Cómo configurar el componente de *middleware* cookie-parser.
- Cómo acceder a las *cookies* en los objetos de petición y de respuesta.
- Cómo configurar la aceptación de *cookies* en varios navegadores webs.

Introducción a las cookies

En 1994, Netscape introdujo el concepto de *cookie* en su navegador web, permitiendo así que los sitios webs almacenaran información particular de sus aplicaciones en los ordenadores de los usuarios. Con el tiempo, las *cookies* se han aceptado e implantado en todos los navegadores webs como Chrome, Edge o Firefox.

Una cookie no es más que un conjunto de datos *inofensivos* almacenado en el navegador del usuario. Las *cookies* se almacenan a petición de los sitios webs. Cada sitio web sólo puede almacenar y acceder a las *cookies* asociadas a ese sitio web; esto quiere decir que un mismo sitio puede tener almacenadas varias *cookies* en el ordenador del usuario y sólo puede acceder a las suyas.

> **Nota.** Es importante recordar que las *cookies* contienen datos *inofensivos*, no son virus, ni *spyware*, ni programas ejecutables, ni *spam*... Son simplemente texto. Debido a la alarma social generada alrededor de las *cookies*, se obliga por Ley, en muchos países de la UE y América, a notificar al usuario que el sitio web hace uso de *cookies*.

Utilidad de las cookies

El objeto principal de las *cookies* es almacenar datos necesarios por la aplicación web en los discos duros de los usuarios. Entre la información que generalmente se almacena, encontramos:

- Identificador de usuario y/o sesión. No se usa datos personales del usuario, pues podría vulnerar la Ley de Protección de Datos, sino una cookie *Pref* que no contiene nada específico del usuario.
- La cesta de la compra. Hay que decir que muchos sitios webs almacenan la cesta de la compra internamente, por ejemplo, en bases de datos, utilizando el identificador del usuario y/o sesión como referencia para identificar la cesta de cada uno de ellos.
- Preferencias del usuario relacionadas, por ejemplo, con el tema de presentación de la página web.

Otra manera de almacenar datos en los navegadores webs es mediante *Web Storage*, estandarizado por el W3C (*World Wide Web Consortium*) responsable de la normalización de otras especificaciones importantes de Internet

como, por ejemplo, HTML5. Esta especificación de almacenamiento la soporta la mayoría de los navegadores webs.

Composición de las cookies

Las *cookies* se transportan o transfieren en los propios mensajes HTTP, mediante campos de cabecera. El servidor transmite sus *cookies* a los clientes mediante el campo de cabecera Set-Cookie; mientras que los clientes utilizan el campo de cabecera Cookie para devolvérselas a los servidores.

Las *cookies* tienen básicamente un nombre, un valor y una lista de parámetros, tal como se muestra a continuación:

```
Nombre=Valor (; Parámetro)*
```

Los distintos parámetros y el valor de la *cookie* se separan entre sí mediante puntos y comas. He aquí un ejemplo de *cookie*:

```
UserId=550e8400-e29b-41d4-a716-44655440000;Expires=Thu, 21 jul 2016 09:33:31
```

Parámetros de las cookies

Un parámetro de cookie (*cookie parameter*) contiene metainformación de la propia *cookie* como, por ejemplo, cuando expira o caduca, a qué dominio pertenece, etc. Se especifican en forma de pares nombre=valor, donde el valor es opcional en algunos parámetros:

```
nombre
nombre[=valor]
```

Básicamente, la lista de parámetros es la siguiente:

- Expires. Indica cuándo caduca la *cookie*, o sea, la fecha a partir de la cual el agente de usuario debe eliminar automáticamente la *cookie* del disco duro del usuario y, por ende, desde la que debe dejar de remitirla al servidor con sus peticiones HTTP.

  El formato de este parámetro es ddd, dd MMM yyyy HH:mm:ss como, por ejemplo, Sat, 09 sep 2013 10:09:00.

  Este parámetro se utiliza para definir el período que el cliente debe retener la *cookie*.

- Domain. Indica el dominio de Internet asociado a la *cookie*, esto es, al que el agente de usuario debe remitir la *cookie* cada vez que le remita una petición HTTP. Así por ejemplo, una *cookie* de google.es no se remitirá bajo ningún concepto a yahoo.com, ni viceversa.

  Este parámetro se utiliza a la hora de definir el ámbito de la *cookie* y su dominio destinatario.

- Path. Indica la ruta del recurso web asociado a la *cookie*.

  Si no se especifica este parámetro, el agente de usuario remitirá la *cookie* siempre que se acceda al dominio, independientemente del recurso web solicitado por el cliente. En cambio, si se indica una ruta, sólo la remitirá cuando solicite ese recurso en cuestión.

  Este parámetro se utiliza, junto con el parámetro domain, para indicar el ámbito de la *cookie*.

- Secure. Indica que el agente de usuario sólo debe remitir o devolver la *cookie* si entre el cliente y el servidor existe una conexión segura como, por ejemplo, una HTTPS.

- HttpOnly. Impide que las aplicaciones clientes puedan acceder a las *cookies*, lo que implica que sólo el servidor pueda modificar la *cookie*. Si no se especifica, la aplicación cliente puede acceder, por ejemplo, mediante JavaScript, al contenido de la cookie.

  Cuando se impide el acceso a la *cookie* a los clientes, se evita los ataques XSS (*Cross-site scripting*). Si un sitio web utiliza las *cookies* para controlar el acceso, un ataque XSS podría permitir a una entidad modificarla y pasar el control de acceso; por esta razón, aquellas *cookies* que se utilicen para pasar directamente el control de acceso deberían estar firmadas y no ser modificables por los clientes.

A continuación, se muestra un ejemplo de *cookie*, la cual usa el servidor para mantener el identificador de la sesión abierta entre cliente y servidor; así, el servidor recibirá con cada mensaje remitido por el cliente, el

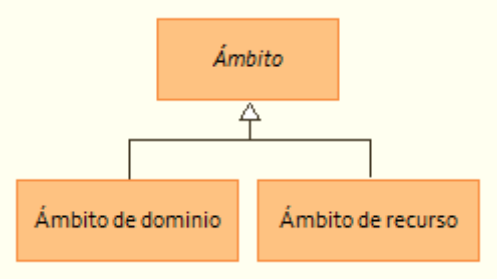
identificador de su sesión:

```
UserId=550e8400-e29b-41d4-a716-44655440000;Expires=Wed, 22 jun 2017 10:26:35;HttpOnly
```

El primer par nombre=valor es el nombre de la *cookie* y su valor. El resto de pares son los parámetros.

Ámbito de las cookies

El ámbito de cookie (*cookie scope*) indica el propietario de la *cookie*, o sea, a qué dominio de Internet se debe remitir. El ámbito lo fija la aplicación web. Se distingue entre ámbito de dominio y ámbito de recurso.



Ámbito de dominio

Cuando sólo se especifica el parámetro domain, se utiliza ámbito de dominio (*domain scope*). Las *cookies* que se encuentran en este ámbito deben ser remitidas por los agentes de usuario con cada mensaje que remitan a ese dominio. Si, por ejemplo, el valor del parámetro domain es google.com, cada vez que se envíe un mensaje HTTP a este dominio habrá que adjuntar la *cookie*.

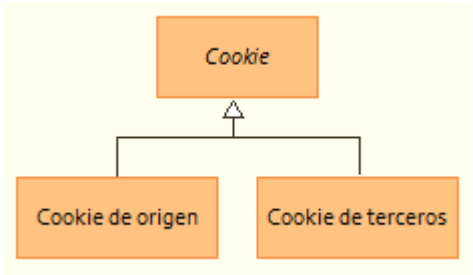
Ámbito de recurso

En cambio, cuando el servidor define tanto domain como path, lo que está haciendo es definir un ámbito de recurso (*resource scope*), limitando así que se remita la *cookie* sólo cuando se acceda a ese recurso o a un recurso que cuelgue de él. Si se trata de un recurso de tipo carpeta, se adjuntará cada vez que se acceda a un recurso ubicado en ella.

Es muy común que tras un mismo dominio haya varias aplicaciones webs, cada una bajo un punto de montaje distinto. Por ejemplo, se puede disponer de dominio.com/app1 y de dominio.com/app2. En estos casos, se recomienda el uso de ámbitos de recurso para que las *cookies* de una aplicación no interfieran ni sean accedidas por las otras.

Destinatario de las cookies

El destinatario de la cookie (*cookie target*) es el servidor web al que el cliente debe remitir la *cookie*. Hasta el momento, hemos indicado que las *cookies* son siempre para la aplicación o sitio web que remite la *cookie* al cliente. Pero también es posible que una aplicación web genere una *cookie* que deba remitirse a otro sitio web, el cual se conoce formalmente como destinatario de la *cookie*. Por esta razón, las *cookies* se pueden clasificar en *cookies* de origen y *cookies* de terceros.



Cookies de origen

Una cookie de origen (*origin cookie*) es aquella en la que el sitio web origen y destino de la *cookie* es el mismo, o sea, el sitio web que generó la *cookie* es el mismo al que el cliente deberá remitirla. La aplicación web define o configura este tipo de *cookies* omitiendo el parámetro domain o bien fijándolo al propio dominio web que aloja la aplicación.

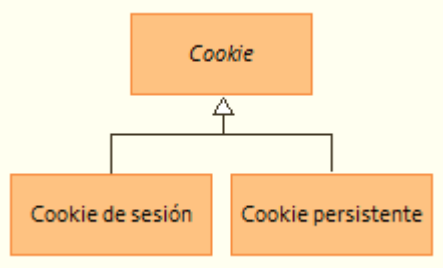
Cookies de terceros

Por su parte, una cookie de tercero (*third-party cookie*) es aquella en la que el sitio web que la genera no es el mismo al que el cliente debe remitirla. La aplicación web origen indicará el dominio destinatario mediante el parámetro domain.

Por seguridad, se recomienda desactivar las *cookies* de terceros en los navegadores webs. De esta manera, se evitará que una aplicación web pueda generar *cookies* de dominios con los que no tiene nada que ver. Pero por desgracia, algunas aplicaciones no funcionan si las cookies de terceros están desactivadas.

Período de retención de las cookies

El período de retención (*retention period*) indica el espacio de tiempo que el cliente debe mantener la *cookie*. Atendiendo a este período, se pueden clasificar en *cookies* de sesión o *cookies* persistentes.



Cookies de sesión

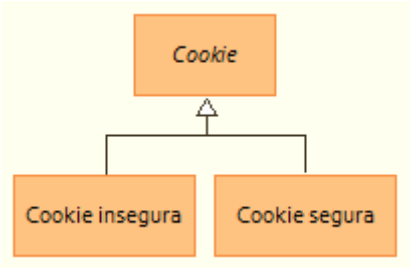
Una cookie de sesión (*session cookie*) es aquella que el navegador debe eliminar cuando se cierre. La aplicación web indica este tipo de *cookie* omitiendo el parámetro expires. Tan pronto como el usuario cierre el navegador, su período de retención se alcanza y ya no se adjuntará en ningún nuevo mensaje remitido con otra nueva sesión del navegador.

Cookies persistentes

En cambio, una cookie persistente (*persistent cookie*) es aquella que presenta fecha de expiración y, por lo tanto, el agente de usuario debe mantenerla en un almacén de datos como, por ejemplo, un archivo en disco, hasta que esta fecha de expiración se alcance. El servidor indica la fecha de expiración mediante el parámetro expires.

Confidencialidad de los datos

Las *cookies* se envían en texto claro o plano, cualquiera que esté monitorizando la red podrá acceder a su contenido. Esto es muy importante tenerlo en cuenta. Atendiendo a cuándo debe remitirse la *cookie*, se distingue entre *cookies* seguras e inseguras.



Cookies inseguras

Una cookie insegura (*unsecure cookie*) es aquella que se transmite siempre que se accede a su destinatario, independientemente de si la conexión abierta entre los extremos es segura o no lo es. Este tipo de *cookies* se configura omitiendo el parámetro secure.

Cookies seguras

Una cookie segura (*secure cookie*) es aquella que sólo se puede transmitir cuando la conexión abierta entre cliente y servidor es segura, por ejemplo, mediante una conexión HTTPS. Si entre cliente y servidor la conexión no es

segura, el cliente no debe remitir la *cookie* en sus peticiones HTTP. Este tipo de *cookies* se marca indicando el parámetro secure.

Cookies firmadas

Una cookie firmada (*signed cookie*) es una *cookie* cuyo valor contiene caracteres adicionales que le sirven al servidor para averiguar si la *cookie* ha sido modificada. Para ello, la aplicación Express se sirve de una clave, que sólo ella debe conocer, conocida formalmente como secreto de cookie (*cookie secret*).

Cuando la información que transmite el servidor en la *cookie* no tiene que ser alterada, se recomienda encarecidamente que se envíe mediante una *cookie* firmada para detectar posibles modificaciones por parte del cliente.

Middleware cookie-parser

El componente de *middleware* cookie-parser se encarga de analizar el campo de cabecera Cookie y facilitar el acceso a las *cookies*, sus valores y sus parámetros, mediante las propiedades cookies y signedCookies del objeto mensaje. Cada *cookie* se representa mediante una propiedad, cuyo nombre es el nombre de la *cookie*; y su valor, su valor.

Para poder utilizar este *middleware*, debemos importar su paquete y registrar su función de *middleware*. No hay que olvidar añadir el paquete cookie-parser a las dependencias de la aplicación, esto es, a la propiedad dependencies del archivo package.json.

Función cookie-parser

La función cookie-parser devuelve una función de *middleware* para analizar los mensajes de petición HTTP y hacer disponible su contenido mediante la propiedad cookies del objeto req:

```
function cookie-parser() : function
function cookie-parser(secret) : function
function cookie-parser(secret, options) : function
```

Parámetro	Tipo de datos	Descripción
secret	string	Clave de firma a usar.
options	object	Opciones del *parser*: • decode (function). Función a usar para descodificar la *cookie*.

Registro de la función de middleware

Recordemos que la función cookie-parser devuelve la función de *middleware* a registrar en la pila de la aplicación. Es por tanto la función devuelta lo que hay que registrar en la aplicación mediante el método use(). Por convenio, el paquete cookie-parser que representa la función homónima se importa como cookieParser. He aquí un ejemplo ilustrativo:

```
import cookieParser from "cookie-parser";
app.use(cookieParser());
```

Este componente debe registrarse en la pila antes de cualquier otro que utilice las propiedades Request.cookies y/o Request.signedCookies.

Objeto request

En una aplicación Express, se puede acceder a las *cookies* remitidas por el cliente mediante las propiedades cookies y signedCookies. Recordemos que el componente analizador de cookies (*cookie parser*) es el encargado de rellenar estas propiedades leyendo el campo de cabecera Cookie de la petición HTTP remitida por el cliente; así pues, no podremos acceder a sus contenidos hasta que este componente haya sido invocado por el motor de *middleware*.

Propiedad request.cookies

Por un lado, tenemos la propiedad de tipo objeto Request.cookies que contiene las *cookies* remitidas por el

cliente en su mensaje HTTP. Cada *cookie* se representa mediante su propia propiedad en el objeto cookies, cuya clave es el nombre de la *cookie* y su valor, su valor.

El siguiente ejemplo muestra cómo listar las distintas *cookies* remitidas en el mensaje de petición HTTP:

```
for (let cookie in req.cookies) {
  console.log(cookie, "->", req.cookies[cookie]);
}
```

Las *cookies* se pueden acceder mediante cualquiera de estas dos formas:

```
req.cookies.nombre
req.cookies["nombre"]
```

Cuando se accede a una *cookie* inexistente, se devuelve undefined. Si el cliente o su agente de usuario no remite ninguna *cookie*, la propiedad contendrá un objeto vacío ({}).

Propiedad request.signedCookies

La propiedad signedCookies permite el acceso a las *cookies* firmadas. Recordemos que éstas son las que el servidor remite al cliente firmadas bajo una clave que sólo él conoce, siendo el *middleware* el encargado de cifrar y descifrar las cookies de manera transparente a la aplicación servidora.

Objeto response

En la sección anterior, hemos presentado cómo acceder a las *cookies* remitidas por el cliente. Ahora, veamos cómo las aplicaciones Express pueden añadir *cookies* a sus respuestas HTTP. Para ello, hay que utilizar el método Response.cookie(). También es posible eliminar *cookies* mediante el método Response.clearCookie().

Método response.cookie()

El método Response.cookie() se encarga de añadir una *cookie* al mensaje de respuesta HTTP. Su signatura es la siguiente:

```
cookie(name, value)
cookie(name, value, options)
```

Parámetro	Tipo de dato	Descripción
name	string	Nombre de la *cookie*.
value	string, object	Valor de la *cookie*. Si se especifica un objeto, se convertirá a formato JSON.
options	Object	Opciones de configuración de la *cookie*: • domain (string): Indica el valor del parámetro domain. Valor predeterminado: dominio de la aplicación. • path (string): Indica el valor del parámetro path. Valor predeterminado: /. • expires (date): Indica el valor del parámetro expires. Valor predeterminado: 0. Lo que implica una *cookie* de sesión. • maxAge (number): Período de expiración en milisegundos. A partir de este valor y la fecha actual, se creará la fecha de expiración, o sea, el valor del parámetro expires. • httpOnly (boolean): Indica el parámetro httpOnly. • secure (boolean): Indica el parámetro secure de la *cookie*. • signed (boolean): Indica si la *cookie* debe firmarse con el valor secret indicado en el *middleware* cookie-parser.

He aquí un ejemplo que ilustra cómo añadir la *cookie* UserId al mensaje de respuesta:

```
res.cookie("UserId", "550e8400-e29b-41d4-a716-44655440000", {maxAge: 900000, secure: false});
```

Recordemos que el servidor remite cookies a los clientes mediante el campo de cabecera Set-Cookie.

Método response.clearCookie()

El método Response.clearCookie() se utiliza para eliminar una *cookie* del mensaje de respuesta:

```
clearCookie(name)
```



```
clearCookie(name, options)
```

Parámetro	Tipo de datos	Descripción
name	string	Nombre de la cookie.
options	object	Opciones de borrado. Las mismas que cookie().

Ejemplo:

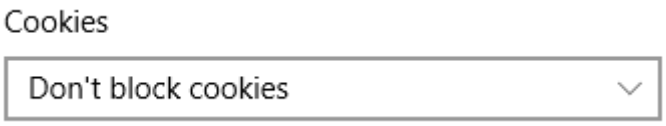
```
res.clearCookie("UserId");
```

Administración de cookies en el cliente

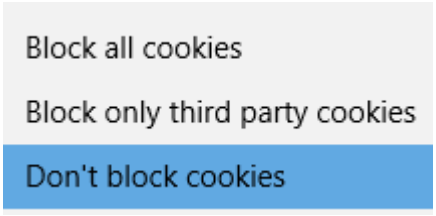
Por lo general, los navegadores webs permiten configurar la administración de *cookies* como, por ejemplo, qué tipo de *cookies* aceptar o dónde almacenarlas. A continuación, se muestra cómo hacerlo en varios de los navegadores más utilizados.

Configuración de Edge

En Edge, la configuración de las *cookies* se encuentra en Settings > Advanced settings > View advanced settings > Cookies:

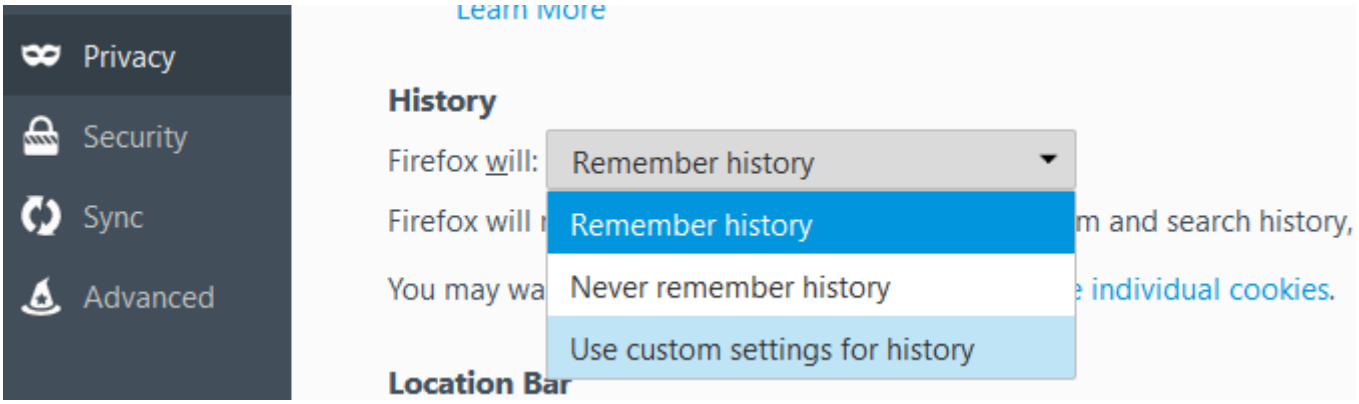


Ahí, podemos configurar si aceptamos *cookies* de origen y/o de terceros:

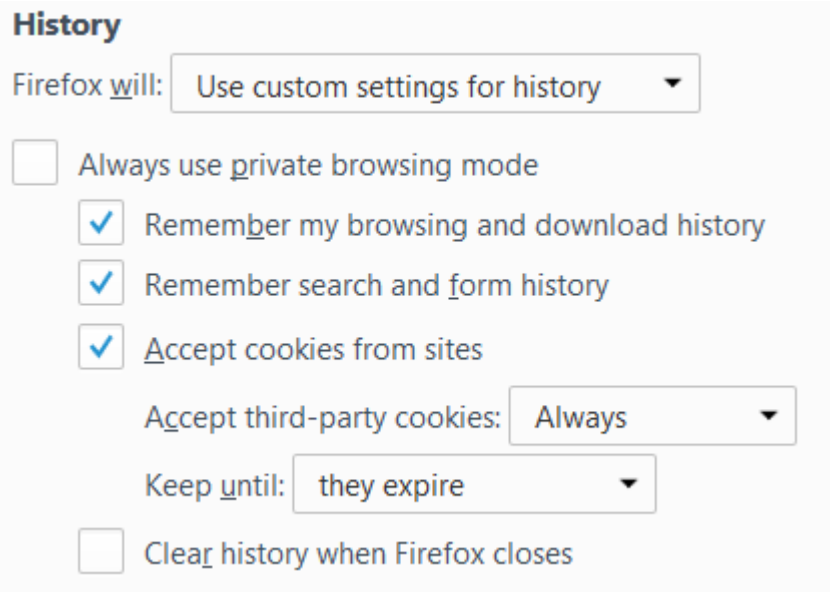


Configuración de Firefox

En Firefox, la configuración de *cookies* se realiza mediante Options > Privacy. A continuación, seleccionar Use custom settings for history en History > Firefox will:



Finalmente, realizar la configuración particular de *cookies*:



Configuración de Chrome

La configuración de cookies en Chrome se realiza mediante Settings > Settings. Hacer clic en Show advanced settings... lo que desplegará más opciones y, a continuación, hacer clic en Content settings...: