Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

HTTP Cookies

Maintaining state with a stateless protocol

Raúl Montes T.

HTTP

- Es un protocolo *stateless*.
- Cada *request/response* HTTP:
  - es independiente de los anteriores y futuros
  - debe contener toda la información necesaria para ser correctamente procesados
- ¿Y cómo se puede mantener una sesión de usuario en la Web entonces?

Sesión de usuario

- Mantener una sesión implica necesariamente mantener un estado (*state*) en algún lugar...
- Tenemos dos opciones para mantener el estado:
  - en el cliente, enviándolo al servidor en cada *request*
  - en el servidor, manteniendo en el cliente un identificador de ese estado que será enviado en cada *request*

Envío de información en un request

- Si almacenamos ya sea un identificador del estado o el estado completo en el cliente, ¿cómo podemos enviarlo al servidor?
- Hemos visto varias opciones:
  - En el path: `/path/to/resource/<more-info>`
  - Como *query parameter*: `/path?more=info`
  - En el *body* del *request* HTTP
- Pero todas ellas requiere que la aplicación Web genere toda la interacción entre cliente y servidor de manera que la información se envíe (ej.: generando todos los *links* con un *query parameter* para el identificador de la sesión)

Cookies al rescate

- Son información en forma de *key-value* que, por instrucción del servidor, el cliente las almacena y automáticamente las envía en cada request.

- Tanto la instrucción de creación por parte del servidor como el envío desde el cliente se realiza mediante HTTP headers.

Response:

```
HTTP/1.1 200 OK
Content-type: text/html
Set-Cookie: session_id=123
```

Siguiente request:

```
GET / HTTP/1.1
Host: www.w3.org
Cookie: session_id=123
```

Detalles

Un header Set-cookie por cada cookie que el servidor quiera enviar

```
Set-Cookie: session_id=123; Expires=Wed, 14 Oct 2015
21:30:00 GMT; Path=/; Secure; HttpOnly
```

- Expires (opcional): la cookie se almacenará hasta la fecha indicada o, si no se envía esta opción, hasta que se cierre el browser (“sesión”).
- Path (opcional): sólo se enviará la cookie cuando el path del request comience con lo indicado aquí (o raíz si no se indica).
- Secure (opcional): si está presente, la cookie sólo se enviará cuando la información viaje encriptada (HTTPS).
- HttpOnly (opcional): la cookie podrá ser accedida sólo en interacciones HTTP. No se podrá acceder por JavaScript en el cliente.

Manteniendo la sesión

- Así, las dos maneras principales en que aplicaciones Web mantienen una sesión de usuario son:
    - El estado se almacena en el servidor, usualmente en una BD, y se envía una cookie con el identificador de esa información de estado al cliente.
    - El estado se almacena completamente en cookies enviadas al cliente.
        - Si queremos almacenar información sensible (como el user_id actualmente identificado), es importante que esta información se almacene de manera segura (encriptada, por ejemplo).

En Rails...

- Rails permite asignar y obtener cookies directamente mediante el método `cookies`.
  - `cookies[:leche] = 'semi descremada'`
  - `cookies[:leche] = { value: 'entera', expires: 1.hour.from_now, httponly: true }`
- Pero para manejar el estado, nos ofrece el método `session`:
  - `session[:user_id] = 123`
- Rails enviará sólo un identificador o el estado completo, y asignará la expiración de las cookies asociadas dependiendo de la configuración de la aplicación (default: información completa se envía al cliente, encriptada)