

DAWE

Web Sockets

Semana 12

LSI Saila

Juanan Pereira <juanan.pereira@ehu.es>

Contenidos

- Introducción a WebSockets
- API WebSockets
- Ejemplos de uso
- Referencias / Bibliografía



Web Sockets: Introducción

- Hemos visto cómo comunicar con el servidor a través de peticiones AJAX (XMLHttpRequest), mediante un modelo de petición/respuesta basado en HTTP
- Cada petición de recurso genera una conexión XMLHttpRequest
- Si el cliente (navegador) quiere conocer el estado de cierto proceso en el servidor, necesita hacer *polling*

Web Sockets: Introducción

- Cabeceras Request: 1356 bytes
- Cabeceras Response: 247 bytes
- Total: más de 1600 bytes (¡por cada petición!)

Request URL: <https://www.google.com/>
Request Method: GET
Status Code: 200 OK
▼ Request Headers [view source](#)
:host: www.google.com
:method: GET
:path: /
:scheme: https
:version: HTTP/1.1
accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
accept-encoding: gzip,deflate,sdch
accept-language: en-US,en;q=0.8
cache-control: no-cache
cookie: PREF=ID=62588ab01a857d9e:U=faf962f3a1ab98eb:FF=0:LD=en:TM=1370625343:LM=1370976257:GM=1:S=HjaEoZj0CWMyahJu; HSID=AK0i1L5lQKhXn5cRA; SSID=Ae91y0Vkl7Mk-Fut9; APISID=3YI866GXqNRRQ0Uw/AyzMG10tZRMh
gKwTD; SAPISID=Qu8br8GtIuvCBSF8/A0ZPV13SQxPddU05C; NID=67=bwA0VN8hATHwdfJDGyoLWR80UtHgCsExuT4nmqKrs6yBJkjJ3ozoZ50FDN76sUdoaezUpPeRuz~sn6x6Lqw2x8WZm2F4t8h7YK_kUisltj9lItdkP6srC0fm893JduyeyV5uB0k5kUcN5u
N_TqDQnU9x9DTD9dc8ea_na8HUORRqjFIDnu8EXHShwbJFTNCG0PTbVuf7t3ifFLVRKX4RoanNRfQhKevcmaw2fpufcJobVR0Kt_Y68BvQaxj-1hqPXC8cIMelwg4tLm43_yiw_id3g; SID=DQAAAMwAAADt5QbrK17dtB00Xau~puD6YAzSHqq5FL2r_NqGECMnJh
F4FTxjm62AcL774JBQYdEg22h0xhA4-u0D_K2IUknP2-at6BAnliBd_W53IBDF486q_iGZPZmB9GP_hkMNUVC3qXL0huJI4EZnojbfJDGi0Q_Ff1b6PQIDzhdASoCLp_Y0LHQhPGd90xmfxDk10fPRKaWi7poYbZSk5DzChRsAWLz0wR_h39iUnHokmGJGmTPZqARmu
W1QAJVayq3Wqk8H4Fcty7uW9qcfooyH_F
pragma: no-cache
user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/28.0.1500.71 Safari/537.36
x-chrome-variations: CMm1yQEIkrbJAQiptskBCMS2yQEIm4TKAQiphcoBCLeFygEIwoXKAQjRhcoB

▼ Response Headers [view source](#)
cache-control: private, max-age=0
content-encoding: gzip
content-type: text/html; charset=UTF-8
date: Thu, 25 Jul 2013 03:51:45 GMT
expires: -1
server: gws
status: 200 OK
version: HTTP/1.1
x-frame-options: SAMEORIGIN
x-xss-protection: 1; mode=block

Web Sockets: Introducción

- Web Sockets es un nuevo API ofertado por HTML5 que te permite mantener una conexión abierta con un servicio web
- Cada vez que haya nuevos datos disponibles, el servicio los enviará al cliente (o clientes), mediante notificaciones con origen en el servidor



Los web sockets permiten abrir conexiones bi-direccionales, full-duplex, entre cliente y servidor

browserquest.mozilla.org

Web Sockets: API

- Crear un web socket:

```
var socket = new WebSocket("ws://tudominio/tuservicio")
```

Web Sockets: API

- Recibir notificaciones y gestionar el evento *open* en cuanto el socket esté abierto contra el servidor:

```
socket.onopen = function() {  
    console.log("Socket conectado!");  
}
```

Web Sockets: API

- Enviar mensajes del cliente al servidor a través del método *send*:

```
socket.send("movimiento válido");
```


Web Sockets: API

- Para recibir mensajes, registramos una función gestora para el evento *onmessage*:

```
socket.onmessage = function(event) {  
    console.log("Respuesta: " + event.data);  
}
```

Para saber más...

Libro HTML5 in Action (Ed. Manning)

(Cap. 4: Messaging: communicating to and from scripts in HTML5)

Making Magic with WebSockets and CSS3

<https://medium.com/outsystems-engineering/making-magic-with-websockets-and-css3-ec22c1dcc8a8>

Dissecting WebSocket's Overhead

<http://tavendo.com/blog/post/dissecting-websocket-overhead/>

Más ejemplos básicos

<http://bohuco.net/blog/2010/07/html5-websockets-example/>

<http://srchea.com/build-a-real-time-application-using-html5-websockets>

<http://www.websocket.org/echo.html>