

```
-rw-----. 1 root root 123 Jul 22 2014 ifcfg-lo
-rw-r--r-- 1 root root 254 Jul 22 2014 ifdown -> ../../../../sbin/ifdown
lrwxrwxrwx 1 root root 20 Jan 29 11:27 ifdown -> ../../../../sbin/ifdown
-rwxr-xr-x 1 root root 627 Jul 22 2014 ifdown-bnep
-rwxr-xr-x 1 root root 511 Jul 22 2014 ifdown-eth
-rwxr-xr-x 1 root root 781 Jul 22 2014 ifdown-etho
-rwxr-xr-x 1 root root 4168 Jul 22 2014 ifdown-ipv6
-rwxr-xr-x 1 root root 11 Jan 29 11:27 ifdown-isdn -> ifdown-ipp
lrwxrwxrwx 1 root root 1481 Jul 22 2014 ifdown-post
-rwxr-xr-x 1 root root 1064 Jul 22 2014 ifdown-ppp
-rwxr-xr-x 1 root root 835 Jul 22 2014 ifdown-routes
-rwxr-xr-x 1 root root 1465 Jul 22 2014 ifdown-sit
-rwxr-xr-x 1 root root 1434 Jul 22 2014 ifdown-tunnel
-rwxr-xr-x 1 root root 18 Jan 29 11:27 ifup -> ../../../../sbin/ifup
lrwxrwxrwx 1 root root 20 Jan 29 11:27 ifup -> ../../../../sbin/ifup
-rwxrwxrwx 1 root root 20 Jan 29 11:27 ifup-aliases
```

DESARROLLO WEB CON MEAN

Curso: DESARROLLO WEB CON MEAN (WEB FULL STACK DEVELOPER)

SOCKET.IO

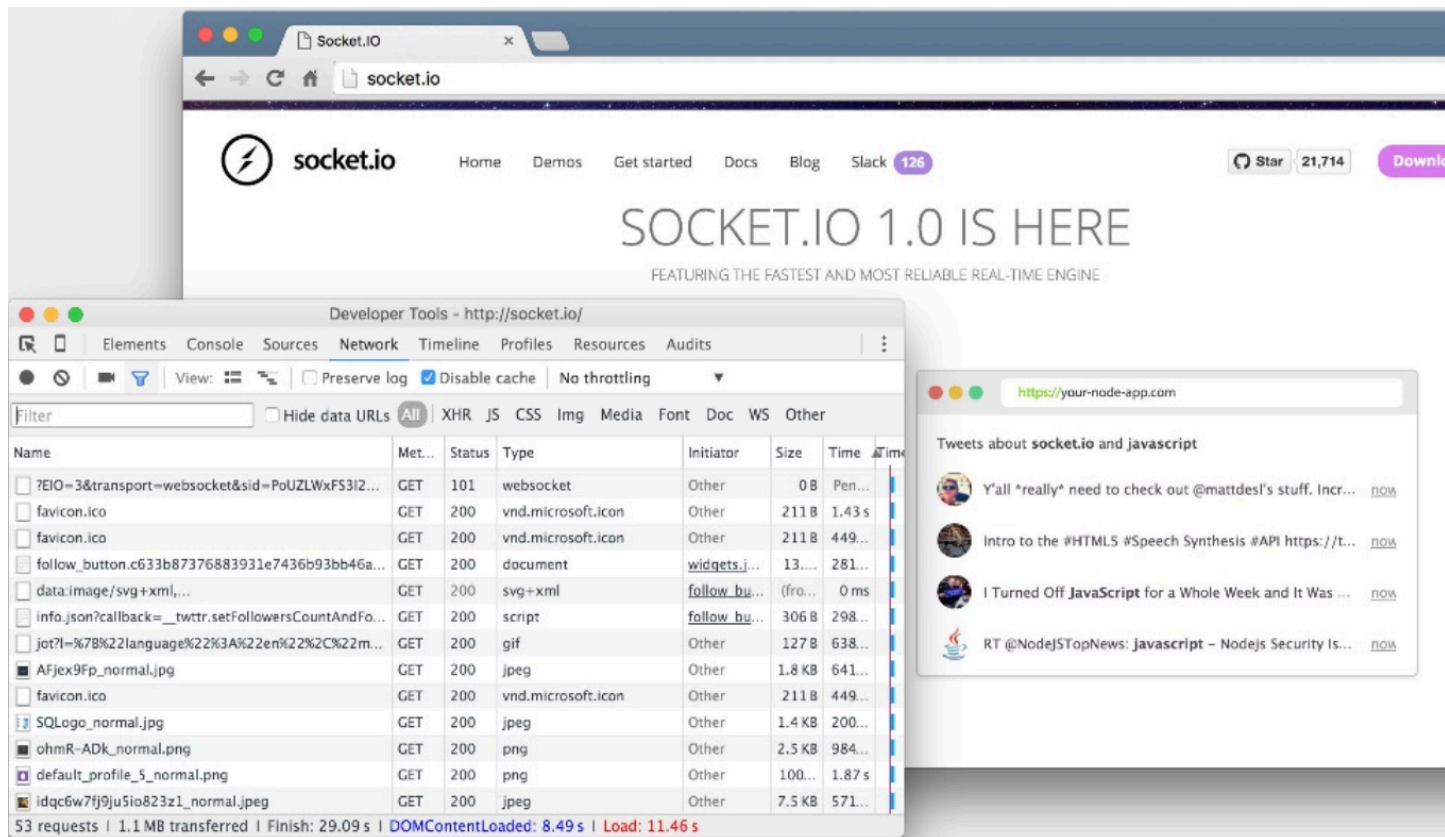
QUE ES SOCKET.IO

- MOTOR EN TIEMPO REAL MUY RÁPIDO
- COMUNICACIÓN DE EVENTOS EN TIEMPO REAL BIDIRECCIONAL
- COMPATIBLE CON DISTINTAS PLATAFORMAS, NAVEGADORES Y DISPOSITIVOS
- USADO PARA
 - MENSAJERÍA INSTANTÁNEA Y CHATS
 - COLABORACIÓN DE DOCUMENTOS
 - STREAMING DE VIDEO/AUDIO/IMAGENES

SOCKET.IO

SOCKET.IO

- CONTENIDO EN EL EXPLORADOR WEB



SOCKET.IO

SOCKET.IO

The image shows a screenshot of the Socket.IO website and its developer tools. The website header includes the Socket.IO logo, navigation links (Home, Demos, Get started, Docs, Blog), a Slack link with 126 members, and a 'One for slack' button. The main heading is 'SOCKET.IO 1.0 IS HERE' with the tagline 'FEATURING THE FASTEST AND MOST RELIABLE REAL-TIME ENGINE'.











The developer tools are open, showing the Network tab. The selected request is a GET request to `http://slack.socket.io/socket.io/?EIO=3&transport=polling&t=1450793936831-0`. The response headers are visible, showing a 200 OK status and various headers including `Access-Control-Allow-Origin: http://socket.io` and `Set-Cookie: io=jg6tHjVhQdLxioJtCjwP`.

On the right, there is a sidebar showing tweets about Socket.IO and JavaScript, including mentions of @mattdeis, #HTML5, #Speech Synthesis, and @NodeJSTopNews.

SOCKET.IO

SOCKET.IO

- SE INICIA CON UN XHR-POLLING (POLLING LARGO)
- CAMBIA A UN TRANSPORTE POR WEBSOCKET
- EN LOS WEBSOCKETS SE VE LA INFORMACIÓN QUE SE ENVÍA

Name	× Headers Preview Response Cookies Timing
 style.css?ver=4.4	<div>▼ General</div> <div>Request URL: http://tweets.socket.io/socket.io/?EIO=3&transport=polling&t=1450793936835-1</div> <div>Request Method: GET</div> <div>Status Code: ● 200 OK</div> <div>Remote Address: 54.241.242.70:80</div> <div>▼ Response Headers view source</div> <div>Access-Control-Allow-Credentials: true</div> <div>Access-Control-Allow-Origin: http://socket.io</div> <div>Connection: keep-alive</div> <div>Content-Length: 101</div> <div>Content-Type: application/octet-stream</div> <div>Date: Tue, 22 Dec 2015 14:20:47 GMT</div> <div>Server: nginx/1.5.11</div> <div>Set-Cookie: io=PoUZLWxFS3I2N6cqAP5J</div> <div>▼ Request Headers view source</div> <div>Accept: */*</div>
 javascripts.js?ver=1.0.0	
 home.js	
 ?EIO=3&transport=polling&t=1450793936831-0	
 ?EIO=3&transport=polling&t=1450793936835-1	
 ?EIO=3&transport=polling&t=1450793937576-2&sid=jg...	
 ?EIO=3&transport=websocket&sid=jg6tHiVhQdLxioJtCjwP	
 ?EIO=3&transport=polling&t=1450793938534-3&sid=Po...	
 ?EIO=3&transport=polling&t=1450793938872-4&sid=Po...	
 ?EIO=3&transport=websocket&sid=PoUZLWxFS3I2N6cqAP5J	
10 / 53 requests 305 KB / 1.1 MB transferred Finish:	

SOCKET.IO

SOCKET.IO A NIVEL DE NAVEGADOR

- EL CLIENTE ESTÁ A LA ESPERA DE NOTIFICACIONES DE EVENTOS Y EMITE EVENTOS.
- CLIENTE A SERVIDOR. SI TOMAMOS COMO EJEMPLO UN CHAT:
 - ESCRITURA, FIN DE ESCRITURA, NUEVO MENSAJE
- SERVIDOR A CLIENTE.
 - ESCRITURA, USUARIO ABANDONA EL CHAT, NUEVO MENSAJE

SOCKET.IO

SOCKET.IO A NIVEL DE NAVEGADOR

- EN CUANTO A LOS MENSAJES ENVIADOS
- CADA EVENTO ESCUCHADO O EMITIDO POSEE UN MENSAJE ASOCIADO
 - CADENAS
 - OBJETOS O ARRAYS
 - BUFFERS
 - BLOBS
 - ARRAYBUFFERS
 - FICHEROS
- ESTOS ÚLTIMOS PERMITEN QUE HAYA IMÁGENES, AUDIO, VIDEO, ETC.
- PARA ANALIZAR DESDE EL NAVEGADOR ESTA COMUNICACIÓN, SE RECOMIENDA:
 - `localStorage.debug="socket.io-client:*`

SOCKET.IO

SOCKET.IO A NIVEL DE SERVIDOR

- NAMESPACES
 - SE ASIGNAN NAMESPACES PARA ASIGNAR ENDPOINTS (PATHS) A SOCKETS
 - PERMITE ASIGNAR DISTINTOS ROLES DE LA MISMA APLICACIÓN
 - POR EJEMPLO, NOTIFICACIONES O CHAT.
- ESPACIO
 - SE PUEDEN DEFINIR CANALES PARA QUE SE PUEDAN UNIR O ABANDONAR
 - POR EJEMPLO CANALES PARA EQUIPOS O CHATS PRIVADOS.

SOCKET.IO

SOCKET.IO

- PARA TRABAJAR CON SOCKETIO, NECESITAMOS USAR UN COMPONENTE WEB. USAREMOS EXPRESS. DECLARAMOS EN EL SERVIDOR DE EXPRESS, SOCKETIO

```
var app = require('express')();  
var server = require('http').Server(app);  
var io = require('socket.io')(server);
```

- Iniciamos el servidor

```
server.listen(8080,function () {  
  var port = server.address().port;  
  console.log('Ejemplo de app escuchando por el puerto', port);  
});
```

SOCKET.IO

SOCKET.IO

- Usamos Express para definir el enrutado

```
app.get('/',function(request,response) {  
  response.sendFile(__dirname + '/index.html');  
});
```

- Declaramos los sockets . Cuando un cliente se conecte, tenemos un socket que emite y escucha eventos.

```
io.on('connection', function (socket) {  
  socket.emit('noticias',{  
    hola: 'mundo'  
  });  
  socket.on('otro evento' function(data){  
    console.log(data);  
  }  
})
```

SOCKET.IO

SOCKET.IO index.html

- Para usar socket.io en el navegador, debemos importar la librería

```
<script src="/socket.io/socket.io.js"/>
```

- Obtenemos en javascript un socket

```
var socket = io.connect('http://localhost:8080');
```

- Cuando el cliente se conecta, se produce un evento connection, y al emitir un mensaje se produce otro evento con el nombre que indiquemos

```
socket.on('noticias', function(data){  
  console.log(data);  
  socket.emit('otro evento',{otro: "evento"});  
});
```

SOCKET.IO

SOCKET.IO

- No es necesario tener navegador para interactuar con socket.io
- Podemos ejecutar bajo node el javascript y ver los resultados si el servidor está levantado.

```
$ node express-ejemplo01/cliente.js  
{otro: 'evento'}
```

SOCKET.IO

SOCKET.IO

- POR OTRA PARTE, SOCKETIO PERMITE
 - EMITIR MENSAJES DE BROADCASTING (A TODOS LOS CLIENTES CONENCTADOS)
 - ENVIA A TODOS LOS CLIENTES CONECTADOS EXCEPTO EL EMISOR
 - ENVIO DE MENSAJES VOLATILES
 - ENVIA Y RECIBE CONFIRMACIONES
 - ESCUCHA OTROS EVENTOS COMO DISCONNECT
 - SE USA PARA COMUNICACIÓN NAVEGADOR/WEB SOCKET

SOCKET.IO

SOCKET.IO

- A

SOCKET.IO

NAMESPACES

- UN NAMESPACE ES UNA FORMA DE SEPARA EL CÓDIGO.
- UN SOCKET POR NAMESPACE

```
var noticias = io('http://localhost:8000/noticias')
```

```
var chat= io('http://localhost:8000/chat')
```

- AUNQUE SE USA LA MISMA CONEXIÓN TCP, SON DOS OBJETOS TOTALMENTE DISTINTOS.
- EL NAMESPACE PRINCIPAL O ROOT SE CREA POR DEFECTO.
- NOTICIAS NO PUEDE MANDAR MENSAJES A CHAT Y VICEVERSA

SOCKET.IO

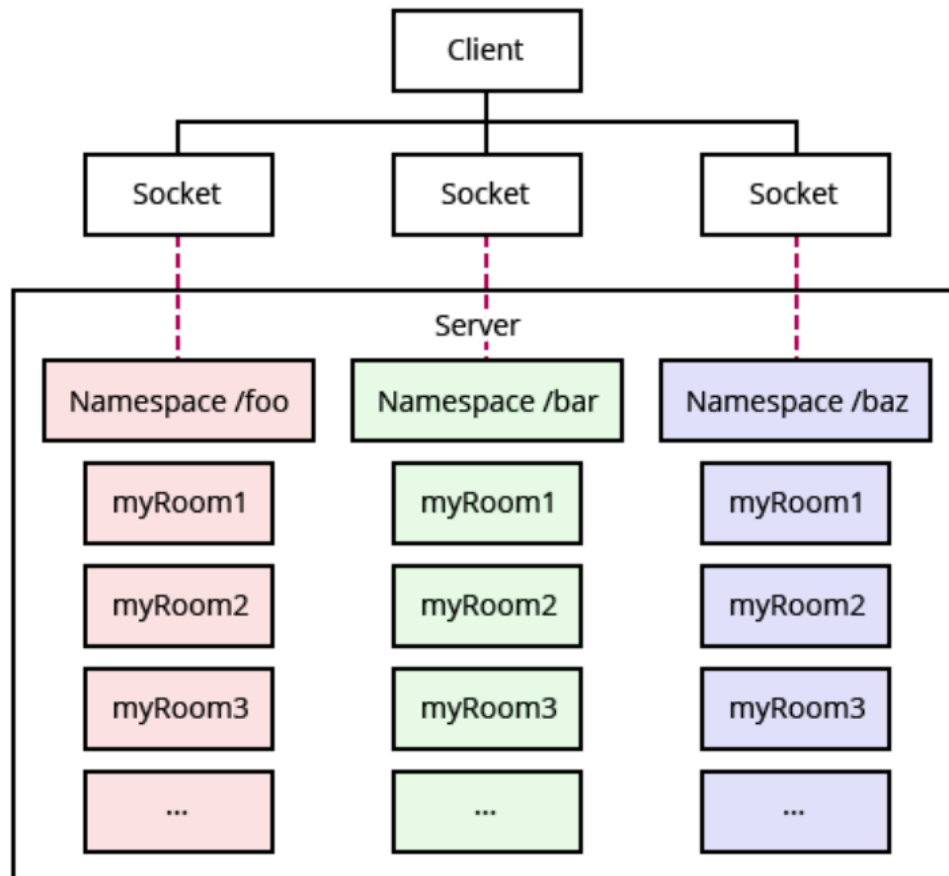
SOCKET.IO

- CUANDO LOS CLIENTES SE CONECTAN A UN NAMESPACE, TODOS RECIBEN LOS EVENTOS PRODUCIDOS EN ELLOS.
- SIN EMBARGO LOS ROOMS SON CONJUNTOS DE SOCKETS, PERMITE ENVIAR MENSAJES A UN SUBGRUPO DE UN NAMESPACE
- SE NECESITA SABER EL NOMBRE DEL ROOM.

SOCKET.IO

SOCKET.IO

- OVERVIEW



SOCKET.IO

SOCKET.IO

- ENVIAR MENSAJES AL NAMESPACE

```
socket.on( 'grito', function( msg ){  
  
    io.emit( 'atodos', msg, socket.id );  
});
```

- En el namespace “grito” se manda un evento “atodos”

SOCKET.IO

SOCKET.IO

- Mandar un mensaje a un ROOM

```
socket.on( 'rooms', function( msg, roomName ) {  
  
    socket.to( nombreRoom).emit( 'heyThere', msg, socket.id );}  
);
```

- Se permite invocar a distintos rooms concatenando .to().to()...

SOCKET.IO

SOCKET.IO

- ENVIAR MENSAJES A TODOS MENOS A TÍ

```
socket.on( 'broadcastMsg', function( msg ) {  
  
    socket.broadcast.emit( 'broadcastFrom', msg, socket.id );  
});
```

- MENSAJES POR BROADCAST A UN ROOM

```
socket.on( 'broadcastMsg', function( msg, roomName ) {  
  
    socket.to( nombreRoom).broadcast.emit( 'broadcastFrom', msg, socket.id );  
});
```

SOCKET.IO

SOCKET.IO

- MANDAR MENSAJE A UN USUARIO

```
socket.on( 'sendToUser', function( msg, userID ){  
  
    var otherSocket = sio.sockets.connected[userID];  
    if( otherSocket ){  
        otherSocket.emit( 'messageFromUser', msg, socket.id );  
    }  
});
```

SOCKET.IO

SOCKET.IO

- SE PERMITE SABER LOS IDS DE USUARIOS

```
var sids = keys( sio.sockets.connected );
```

- Y LOS IDS DE UN ROOM

```
var socketsInRoom = sio.sockets.adapter.rooms[roomName];  
var sids = ( socketsInRoom ) ? keys( socketsInRoom ) : []
```

- PARA UN CHAT, ES SENCILLO ALMACENAR UN DICCIONARIO CON LOS IDS Y LOS APODOS DE LOS CLIENTES
- DE ESA MANERA PODEMOS COMUNICARNOS VIA APODO CON ELLOS.

SOCKET.IO

EJERCICIO

- REALICE UN CHAT QUE PERMITA CONECTAR DE FORMA PÚBLICA CON TODO EL MUNDO
- DEBE PODER CREAR SALAS DE CHAT INDEPENDIENTES DONDE UNOS POCOS PUEDAN CONVERSAR FUERA DE LA SALA PRINCIPAL POR MEDIO DE UN NOMBRE
- CREE TAMBIÉN LA OPCIÓN DE MENSAJES PRIVADOS, NO SALAS DE CHAT DE DOS USUARIOS.