Advanced HTML5

Web Sockets

Craig Shoemaker

http://craigshoemaker.net

@craigshoemaker

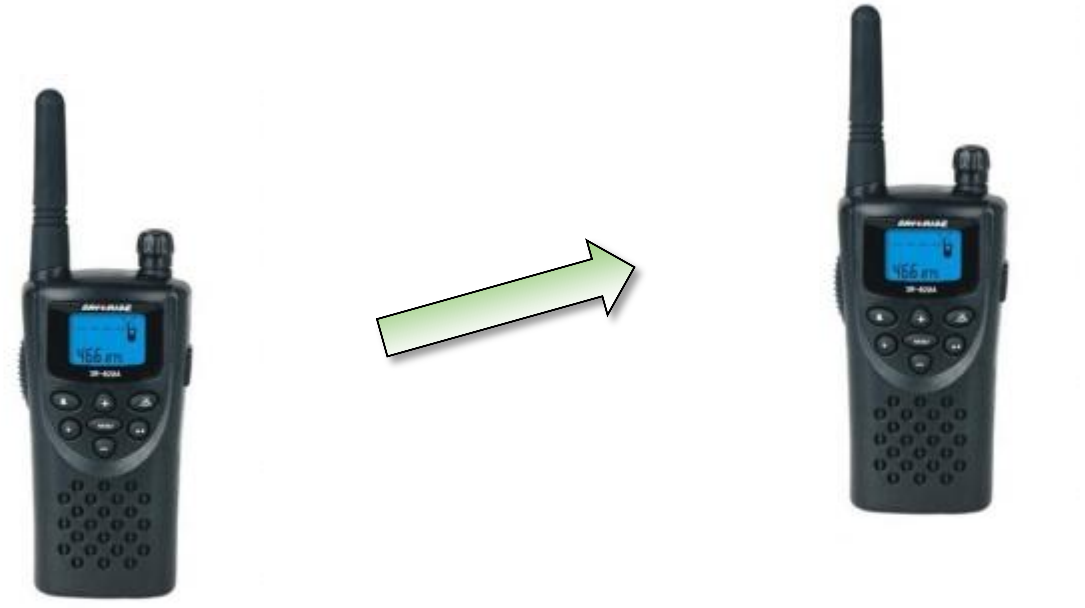
Outline

- Introduction
- What are Web Sockets?
- Communication Systems
- Real-Time Data Before Web Sockets
- Advantages of Web Sockets
- Web Sockets vs. HTTP
- Web Socket Server Anatomy
- Demos
  - Review Nugget Server Implementation
  - Connecting to Server
  - Sending Messages to and From the Server
  - Stock Ticker

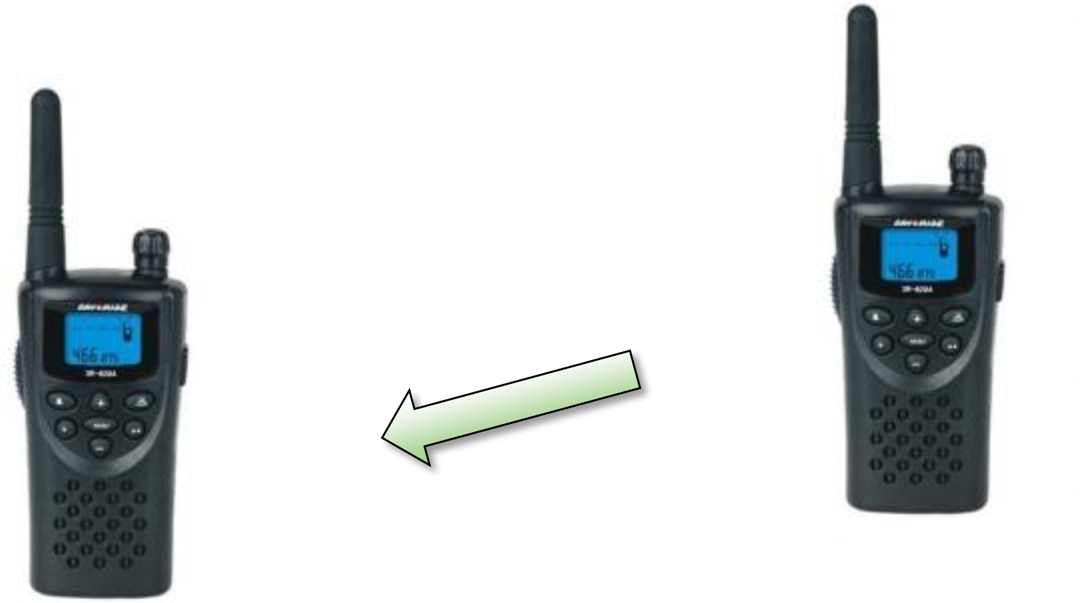
What Are Web Sockets?

Bidirectional,
full duplex client/server
communication

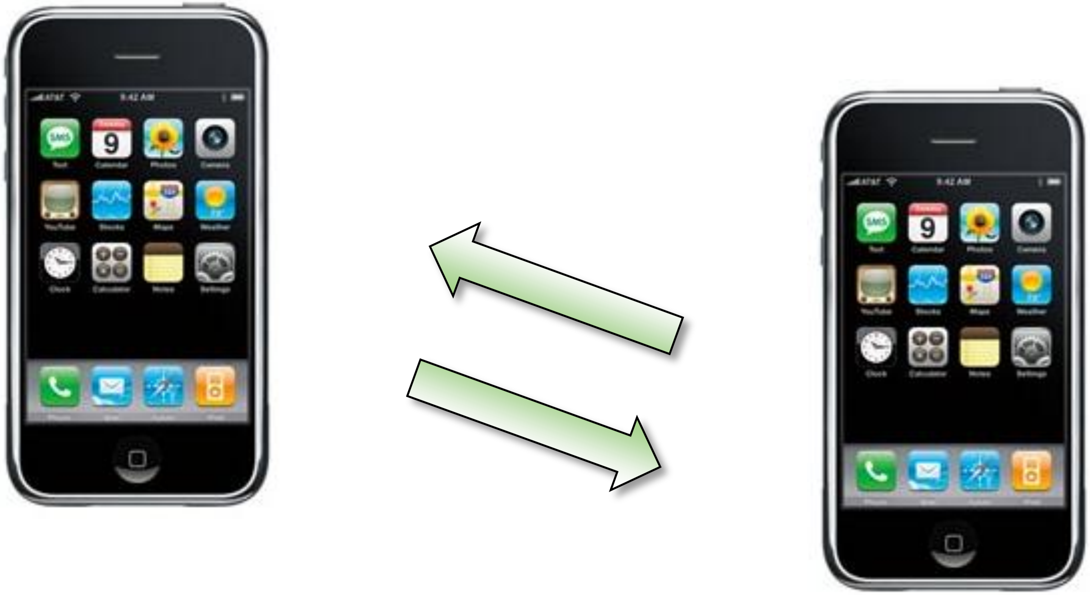
Half Duplex Communications



Half Duplex Communications



Full Duplex Communications

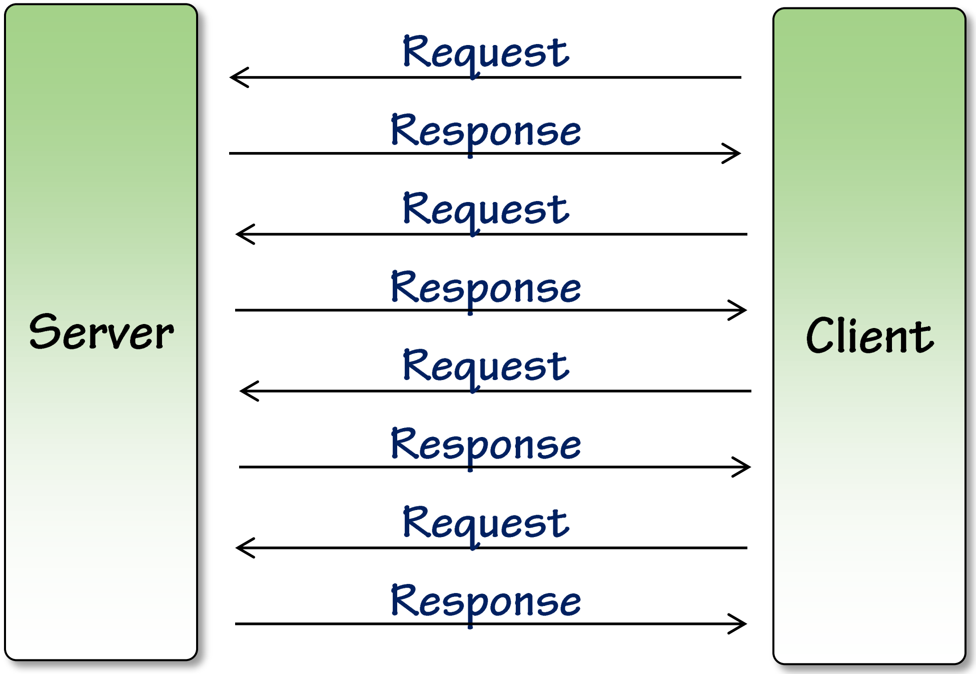


Real Time Data Before Web Sockets

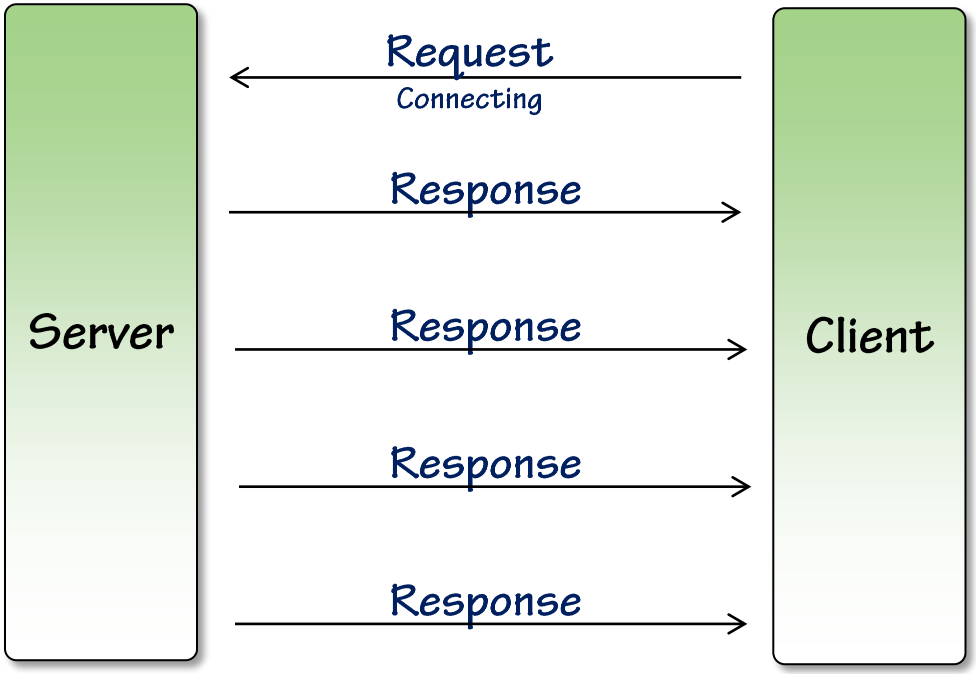
Real Time Data Before Web Sockets

Polling, Long Polling & Streaming

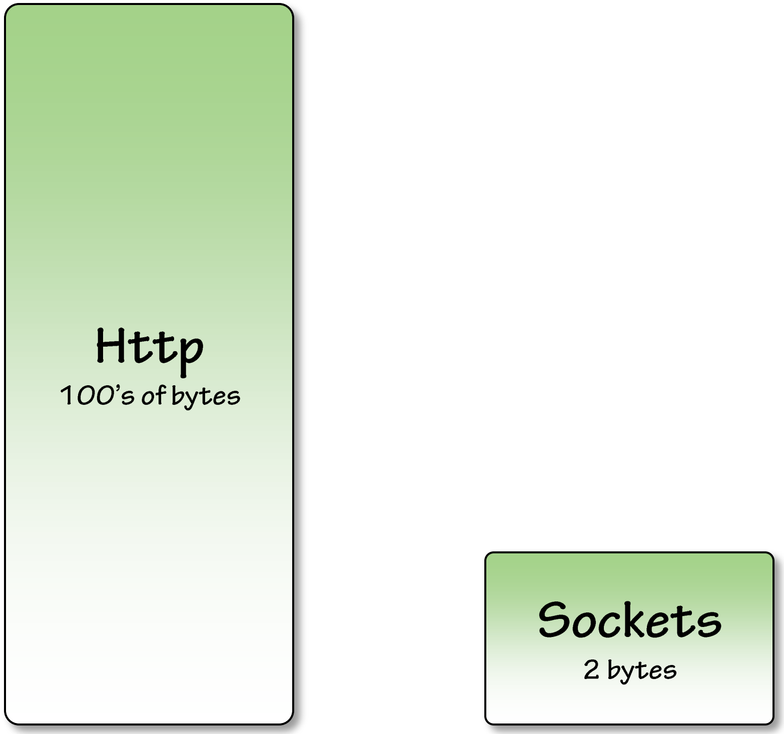
Polling


Server
Request
Response
Request
Response
Request
Response
Request
Response
Client

Sockets


Request
Connecting
Response
Response
Response
Response
Server
Client

HTTP Header vs. Socket Header



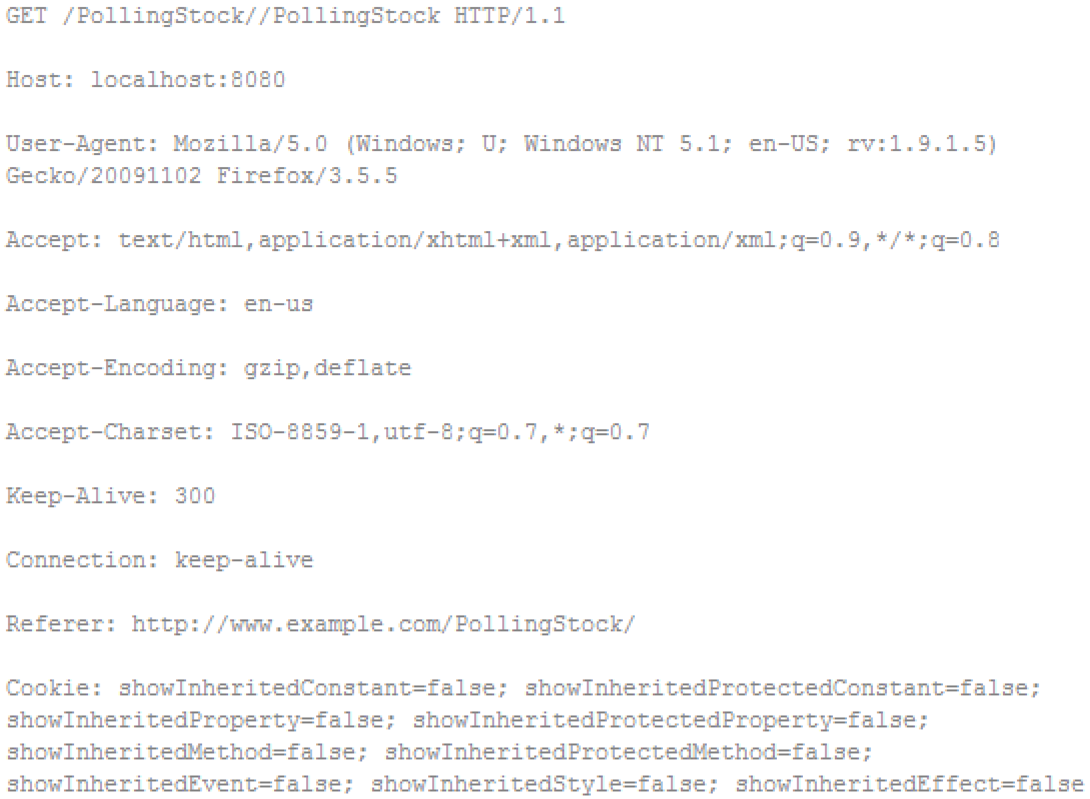
Http

100's of bytes

Sockets

2 bytes

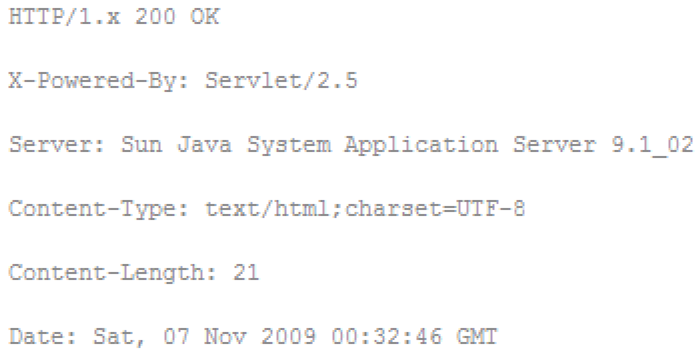
HTTP Request



```
GET /PollingStock//PollingStock HTTP/1.1

Host: localhost:8080

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.5)
Gecko/20091102 Firefox/3.5.5

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-us

Accept-Encoding: gzip,deflate

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7

Keep-Alive: 300

Connection: keep-alive

Referer: http://www.example.com/PollingStock/

Cookie: showInheritedConstant=false; showInheritedProtectedConstant=false;
showInheritedProperty=false; showInheritedProtectedProperty=false;
showInheritedMethod=false; showInheritedProtectedMethod=false;
showInheritedEvent=false; showInheritedStyle=false; showInheritedEffect=false
```

Source: http://websocket.org/quantum.html

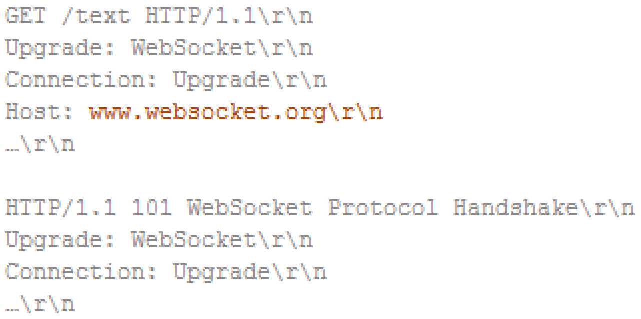
HTTP Response



```
HTTP/1.x 200 OK

X-Powered-By: Servlet/2.5

Server: Sun Java System Application Server 9.1_02

Content-Type: text/html;charset=UTF-8

Content-Length: 21

Date: Sat, 07 Nov 2009 00:32:46 GMT
```

Source: http://websocket.org/quantum.html

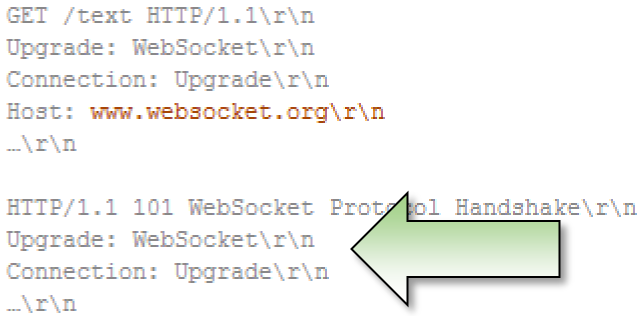
Upgrade from HTTP to Web Socket Protocol



```
GET /text HTTP/1.1\r\n
Upgrade: WebSocket\r\n
Connection: Upgrade\r\n
Host: www.websocket.org\r\n
…\r\n

HTTP/1.1 101 WebSocket Protocol Handshake\r\n
Upgrade: WebSocket\r\n
Connection: Upgrade\r\n
…\r\n
```

Source: http://websocket.org/quantum.html

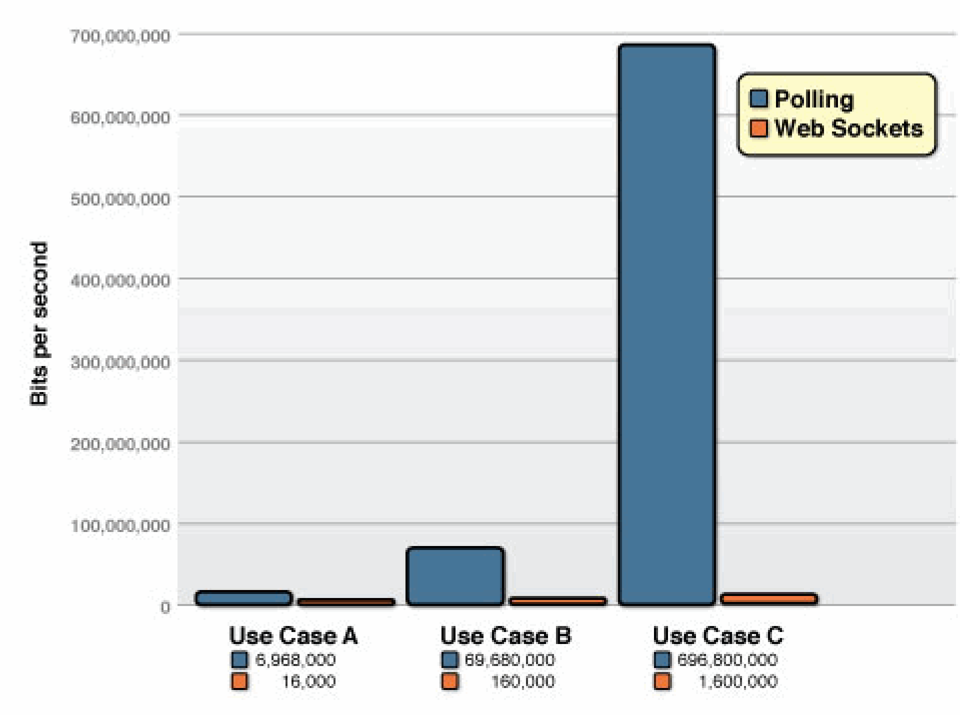
Upgrade from HTTP to Web Socket Protocol



```
GET /text HTTP/1.1\r\n
Upgrade: WebSocket\r\n
Connection: Upgrade\r\n
Host: www.websocket.org\r\n
…\r\n

HTTP/1.1 101 WebSocket Protocol Handshake\r\n
Upgrade: WebSocket\r\n
Connection: Upgrade\r\n
…\r\n
```

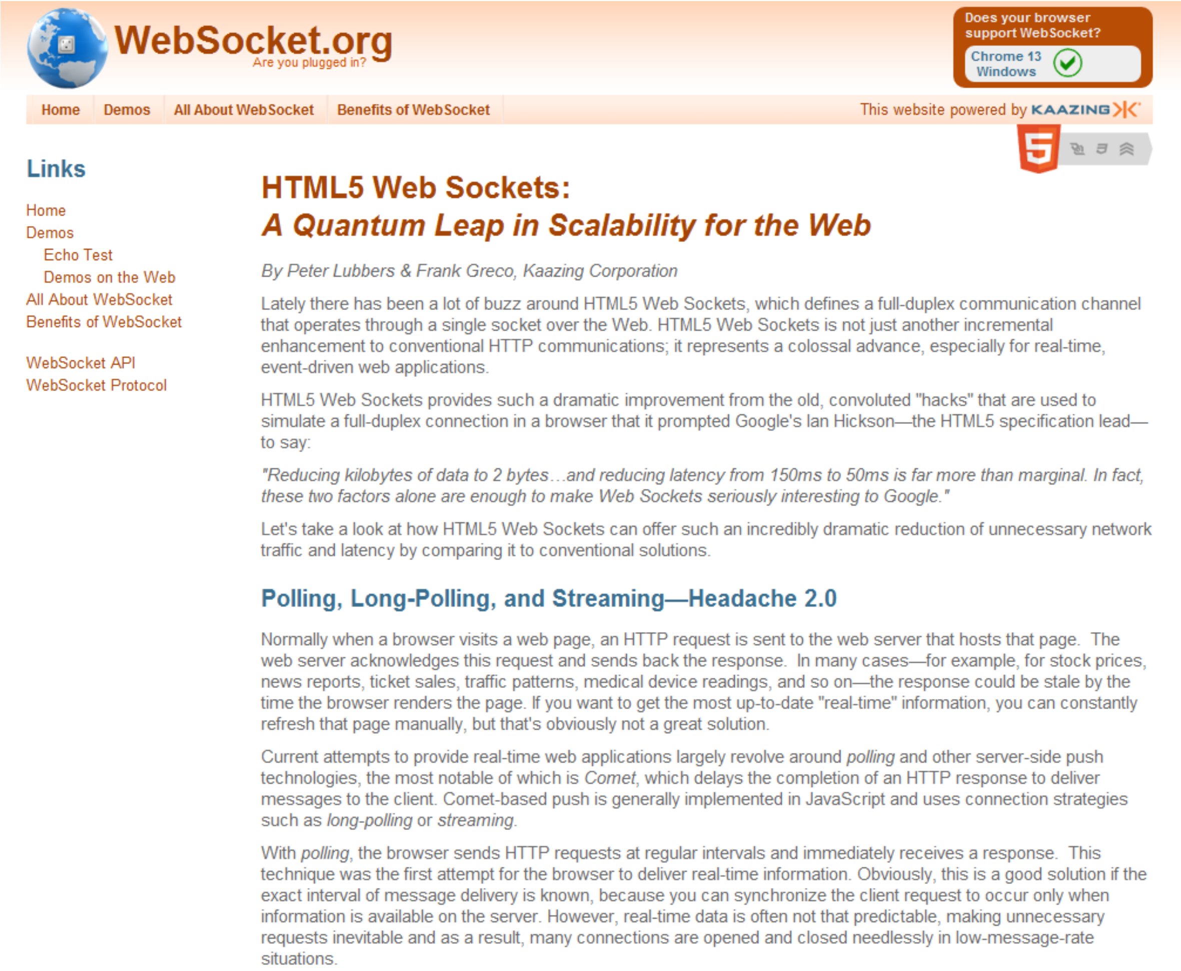
Source: http://websocket.org/quantum.html

HTTP Header vs. Socket Header



Source: http://websocket.org/quantum.html

http://websocket.org/quantum.html



Links

Home
Demos
Echo Test
Demos on the Web
All About WebSocket
Benefits of WebSocket

WebSocket API
WebSocket Protocol

HTML5 Web Sockets:
A Quantum Leap in Scalability for the Web

By Peter Lubbers & Frank Greco, Kaazing Corporation

Lately there has been a lot of buzz around HTML5 Web Sockets, which defines a full-duplex communication channel that operates through a single socket over the Web. HTML5 Web Sockets is not just another incremental enhancement to conventional HTTP communications; it represents a colossal advance, especially for real-time, event-driven web applications.

HTML5 Web Sockets provides such a dramatic improvement from the old, convoluted "hacks" that are used to simulate a full-duplex connection in a browser that it prompted Google's Ian Hickson—the HTML5 specification lead—to say:

"Reducing kilobytes of data to 2 bytes…and reducing latency from 150ms to 50ms is far more than marginal. In fact, these two factors alone are enough to make Web Sockets seriously interesting to Google."

Let's take a look at how HTML5 Web Sockets can offer such an incredibly dramatic reduction of unnecessary network traffic and latency by comparing it to conventional solutions.

Polling, Long-Polling, and Streaming—Headache 2.0

Normally when a browser visits a web page, an HTTP request is sent to the web server that hosts that page. The web server acknowledges this request and sends back the response. In many cases—for example, for stock prices, news reports, ticket sales, traffic patterns, medical device readings, and so on—the response could be stale by the time the browser renders the page. If you want to get the most up-to-date "real-time" information, you can constantly refresh that page manually, but that's obviously not a great solution.

Current attempts to provide real-time web applications largely revolve around *polling* and other server-side push technologies, the most notable of which is *Comet*, which delays the completion of an HTTP response to deliver messages to the client. Comet-based push is generally implemented in JavaScript and uses connection strategies such as *long-polling* or *streaming*.

With *polling*, the browser sends HTTP requests at regular intervals and immediately receives a response. This technique was the first attempt for the browser to deliver real-time information. Obviously, this is a good solution if the exact interval of message delivery is known, because you can synchronize the client request to occur only when information is available on the server. However, real-time data is often not that predictable, making unnecessary requests inevitable and as a result, many connections are opened and closed needlessly in low-message-rate situations.

Browser Support

Chrome
4.0+

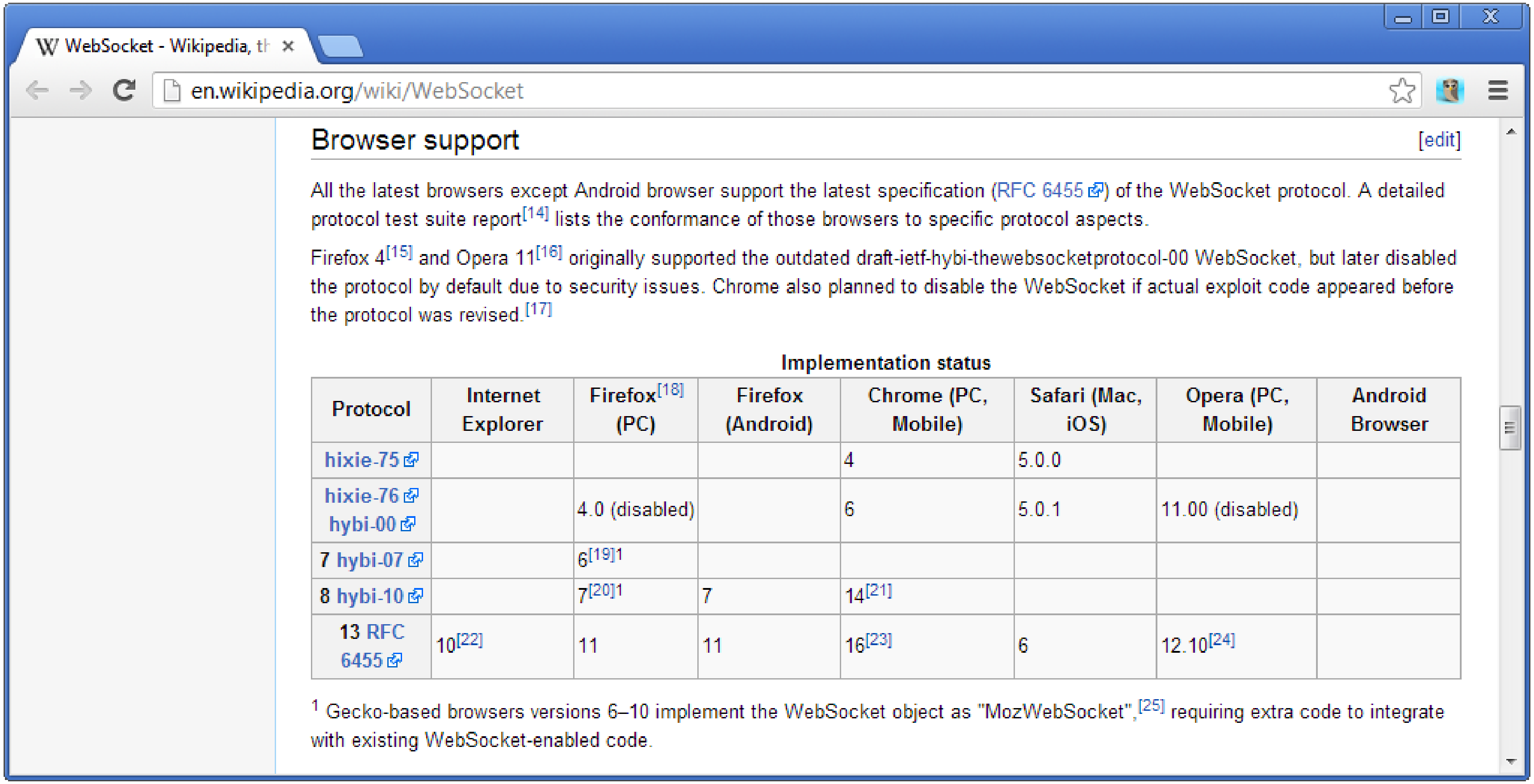
Firefox
4+
(partial)

Opera
11.0+
(partial)

Safari iOS & Mac
5.0+

IE
10+

Web Socket Protocols



Browser support

[edit]

All the latest browsers except Android browser support the latest specification (RFC 6455) of the WebSocket protocol. A detailed protocol test suite report[14] lists the conformance of those browsers to specific protocol aspects.

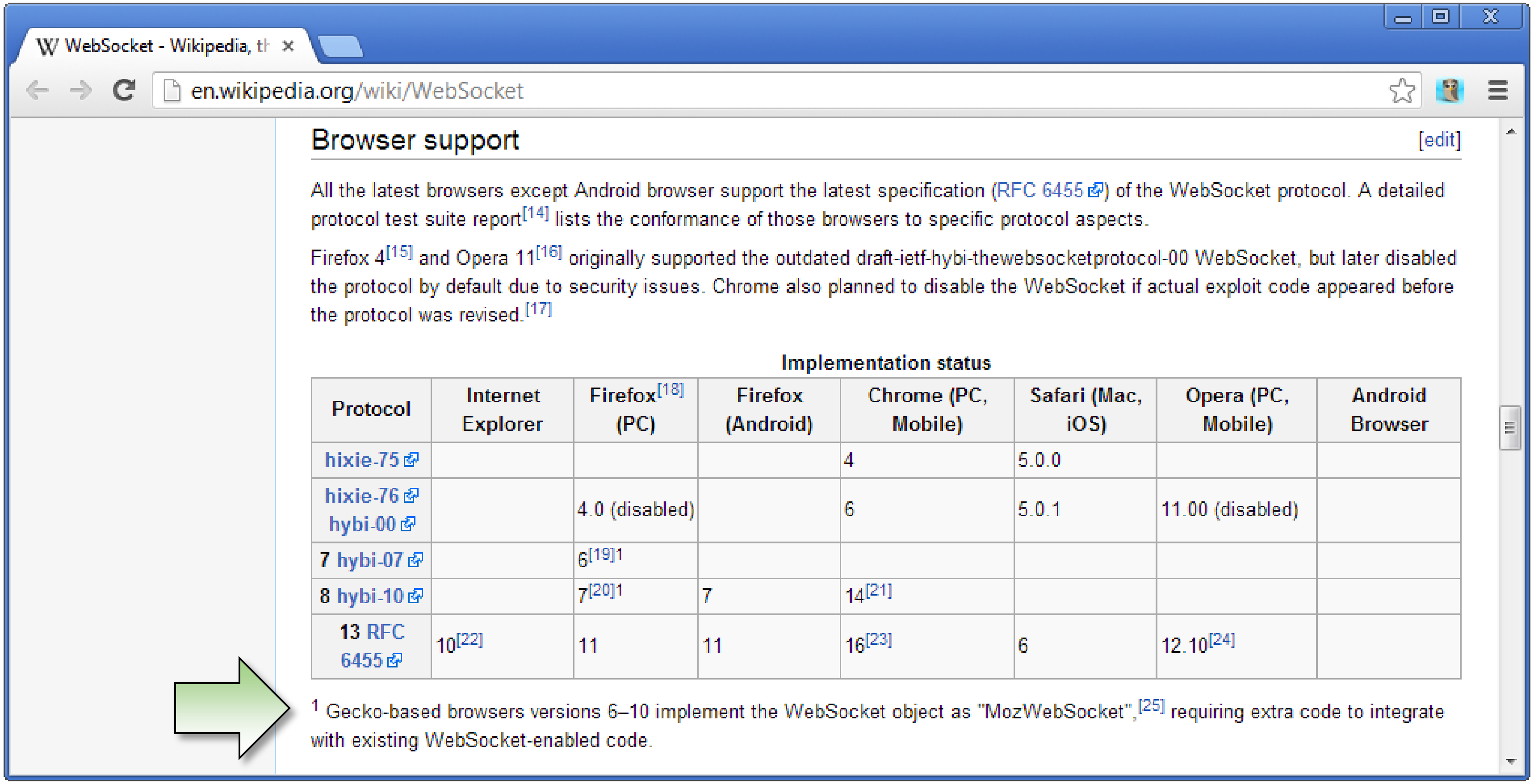
Firefox 4[15] and Opera 11[16] originally supported the outdated draft-ietf-hybi-thewebsocketprotocol-00 WebSocket, but later disabled the protocol by default due to security issues. Chrome also planned to disable the WebSocket if actual exploit code appeared before the protocol was revised.[17]

Implementation status

Protocol	Internet Explorer	Firefox[18] (PC)	Firefox (Android)	Chrome (PC, Mobile)	Safari (Mac, iOS)	Opera (PC, Mobile)	Android Browser
hixie-75				4	5.0.0		
hixie-76 hybi-00		4.0 (disabled)		6	5.0.1	11.00 (disabled)	
7 hybi-07		6[19][1]					
8 hybi-10		7[20][1]	7	14[21]			
13 RFC 6455	10[22]	11	11	16[23]	6	12.10[24]	

[1] Gecko-based browsers versions 6–10 implement the WebSocket object as "MozWebSocket",[25] requiring extra code to integrate with existing WebSocket-enabled code.

Web Socket Protocols


W WebSocket - Wikipedia, th ×
en.wikipedia.org/wiki/WebSocket
[edit]

Browser support

All the latest browsers except Android browser support the latest specification (RFC 6455) of the WebSocket protocol. A detailed protocol test suite report[14] lists the conformance of those browsers to specific protocol aspects.

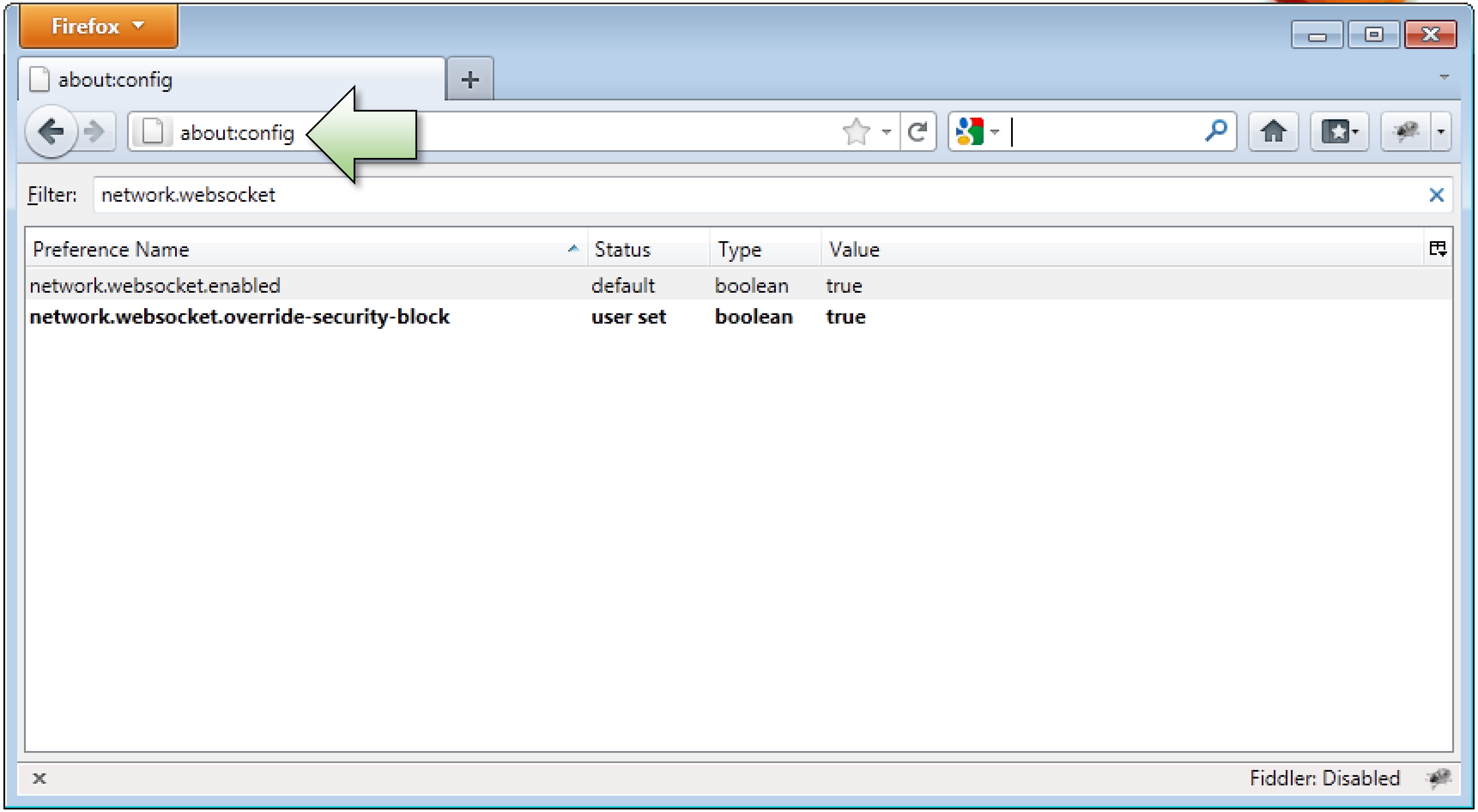
Firefox 4[15] and Opera 11[16] originally supported the outdated draft-ietf-hybi-thewebsocketprotocol-00 WebSocket, but later disabled the protocol by default due to security issues. Chrome also planned to disable the WebSocket if actual exploit code appeared before the protocol was revised.[17]

Implementation status

Protocol	Internet Explorer	Firefox[18] (PC)	Firefox (Android)	Chrome (PC, Mobile)	Safari (Mac, iOS)	Opera (PC, Mobile)	Android Browser
hixie-75				4	5.0.0		
hixie-76 hybi-00		4.0 (disabled)		6	5.0.1	11.00 (disabled)	
7 hybi-07		6[19]1					
8 hybi-10		7[20]1	7	14[21]			
13 RFC 6455	10[22]	11	11	16[23]	6	12.10[24]	

[1] Gecko-based browsers versions 6–10 implement the WebSocket object as "MozWebSocket",[25] requiring extra code to integrate with existing WebSocket-enabled code.

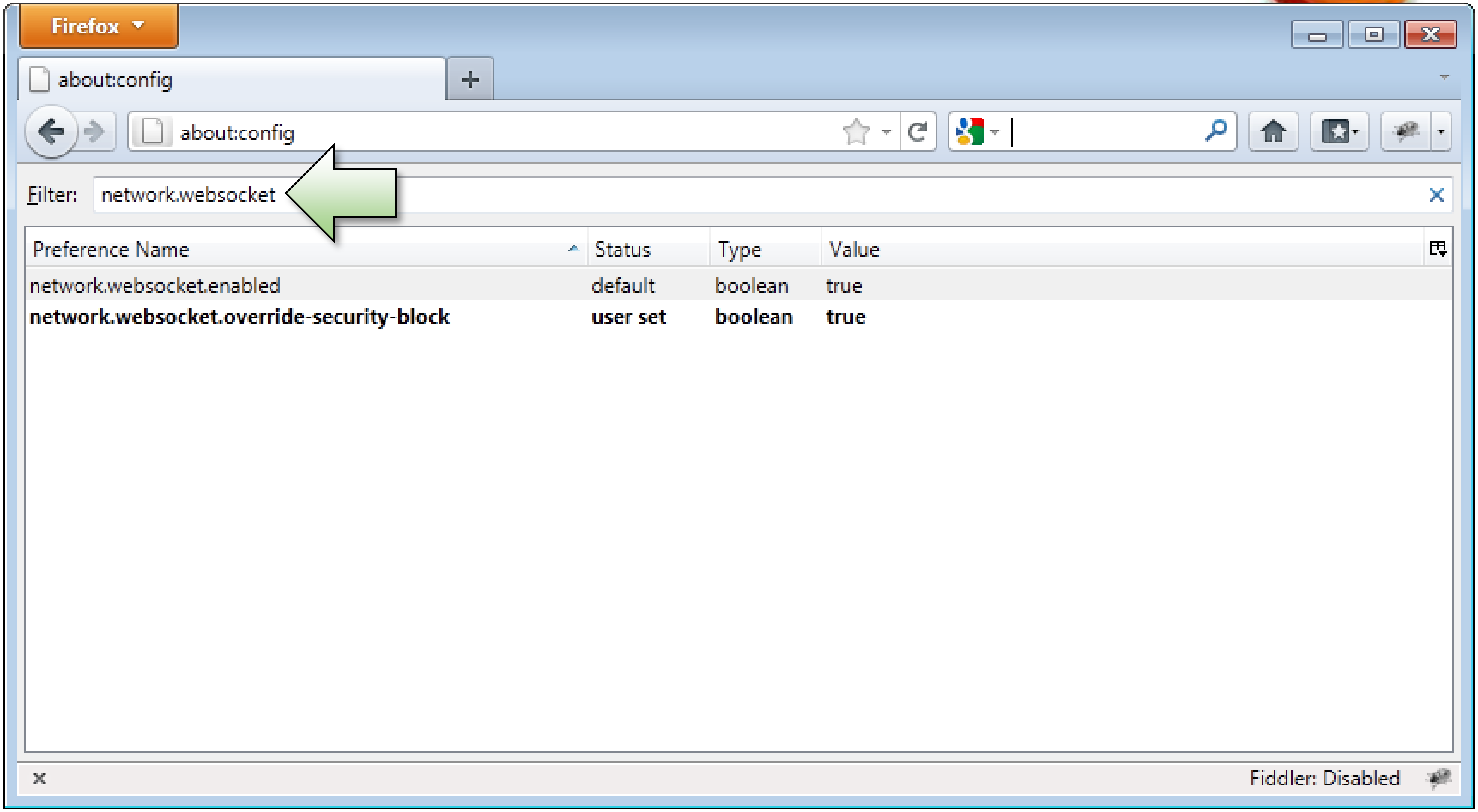
Enabling Web Sockets in FireFox



Filter: network.websocket

Preference Name	Status	Type	Value
network.websocket.enabled	default	boolean	true
network.websocket.override-security-block	**user set**	**boolean**	**true**

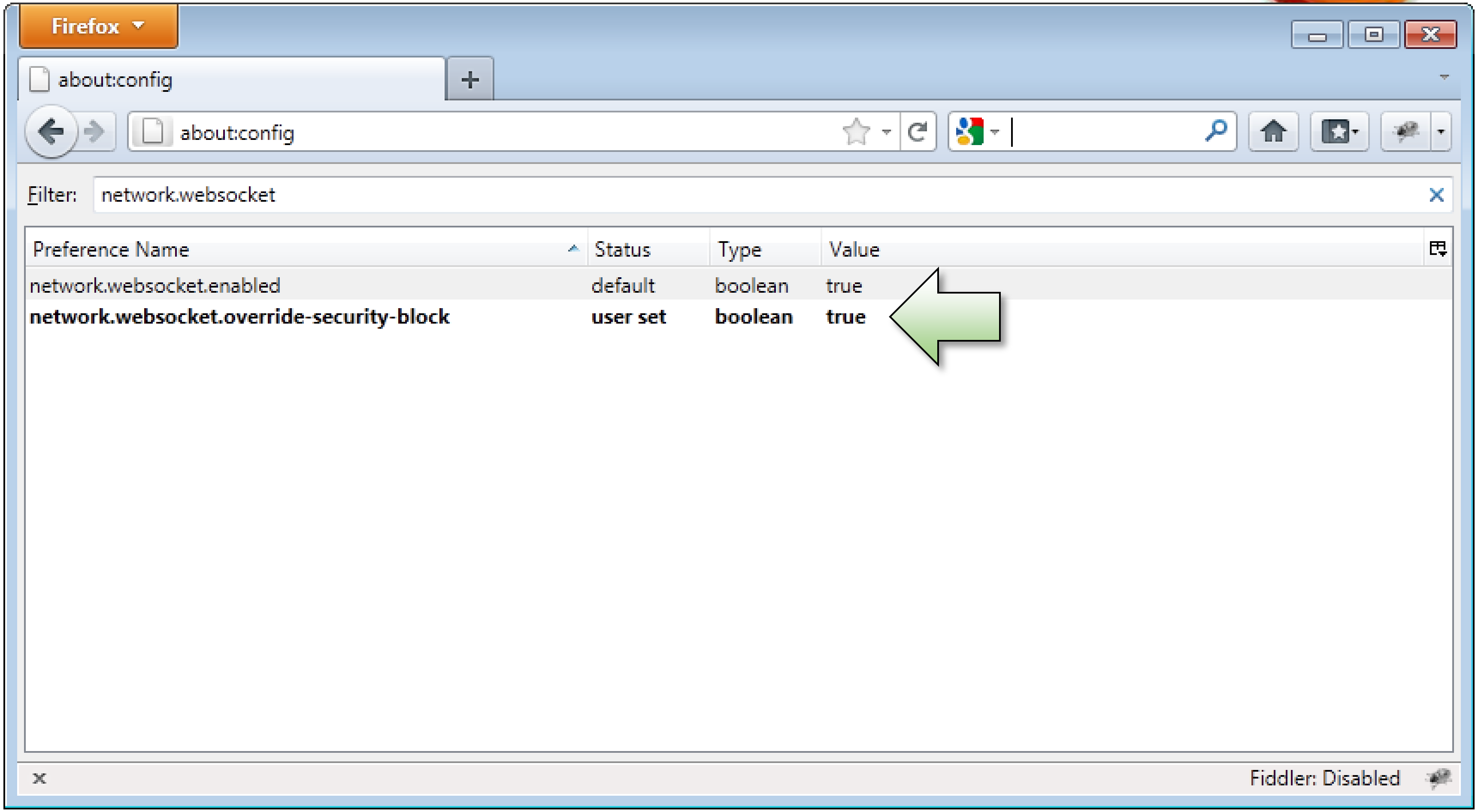
Enabling Web Sockets in FireFox



Filter: network.websocket

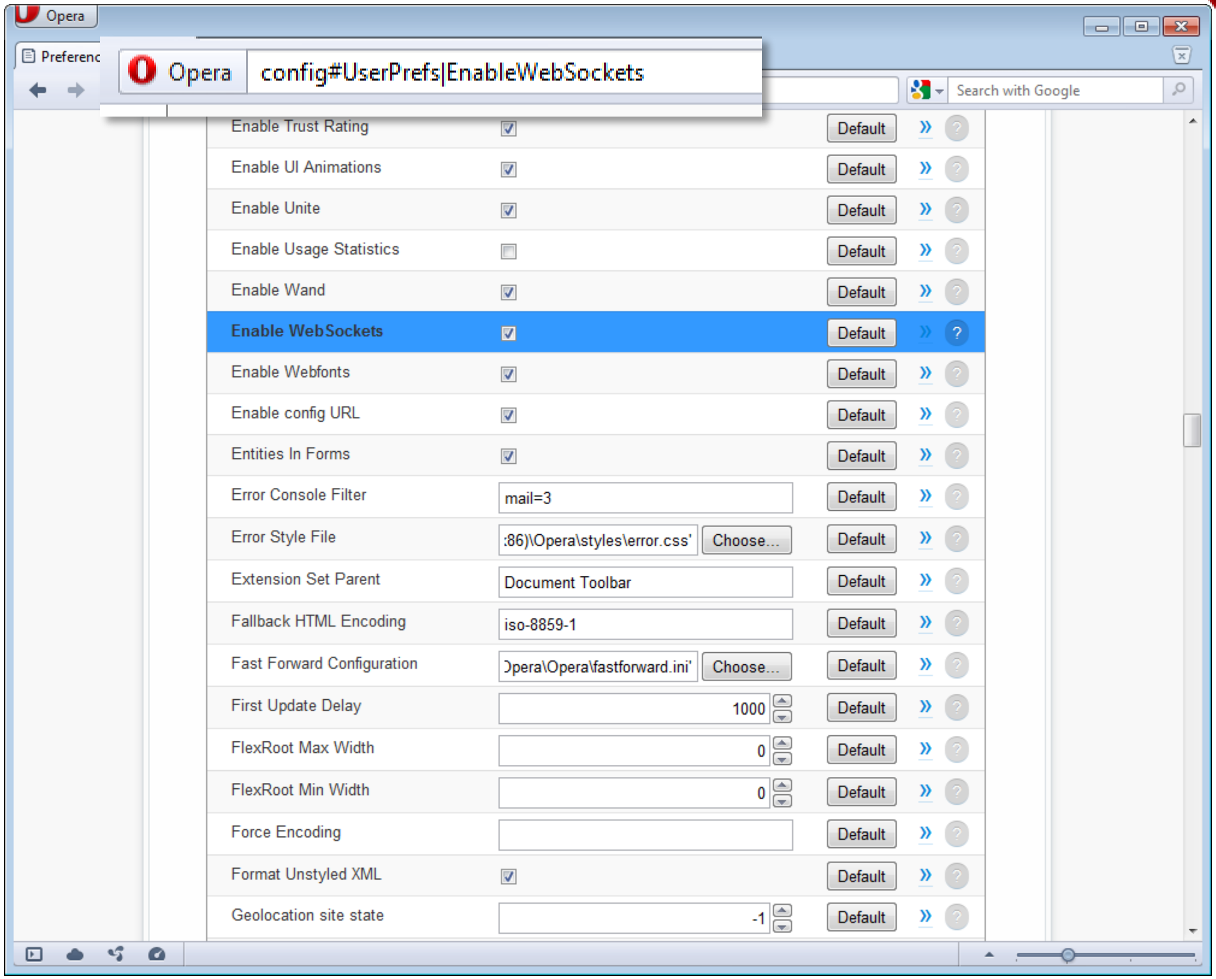
Preference Name	Status	Type	Value
network.websocket.enabled	default	boolean	true
network.websocket.override-security-block	**user set**	**boolean**	**true**

Enabling Web Sockets in FireFox



Preference Name	Status	Type	Value
network.websocket.enabled	default	boolean	true
network.websocket.override-security-block	**user set**	**boolean**	**true**

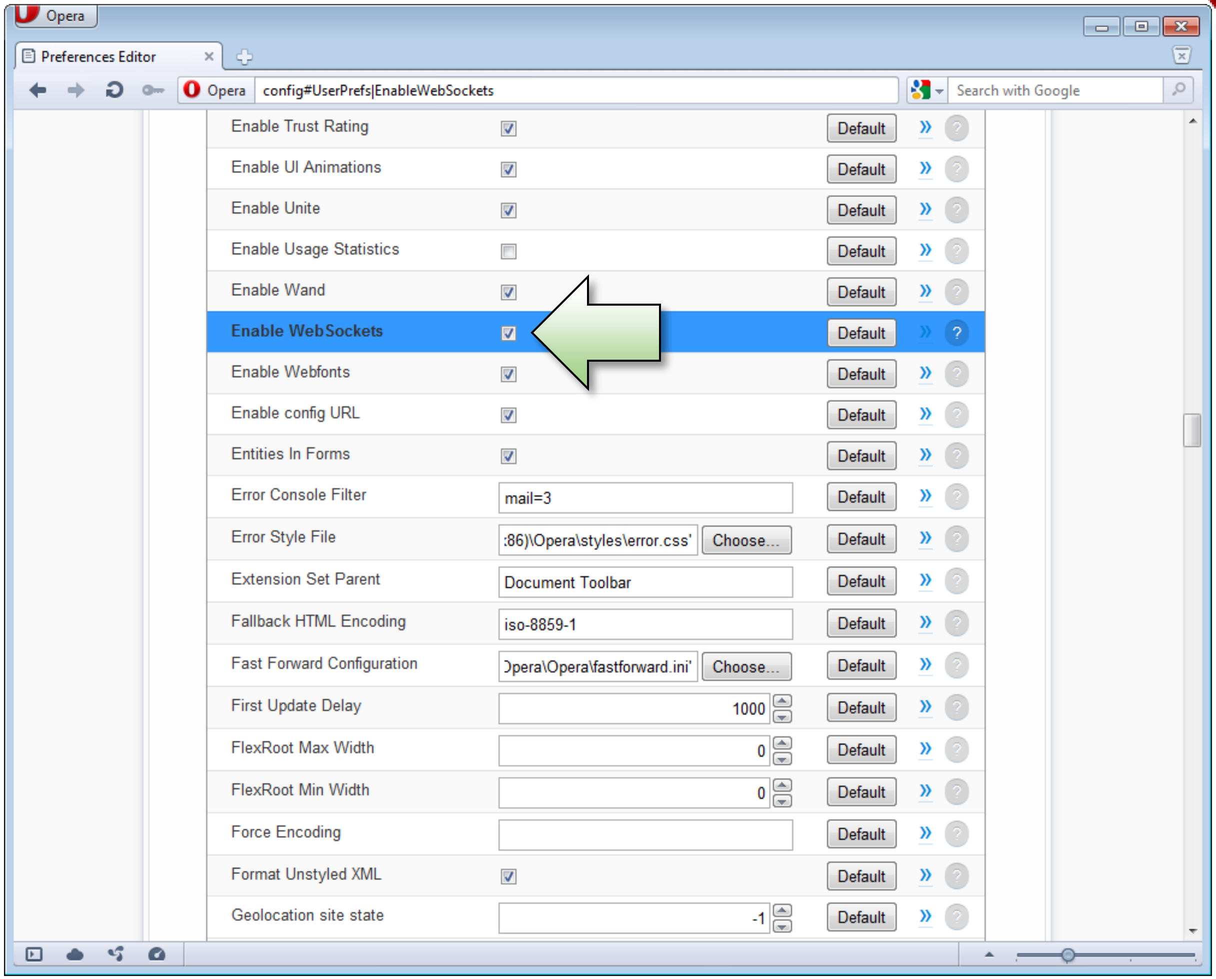
Enabling Web Sockets in Opera



Opera config#UserPrefs|EnableWebSockets

Enable Trust Rating	☑	Default	» ?
Enable UI Animations	☑	Default	» ?
Enable Unite	☑	Default	» ?
Enable Usage Statistics	□	Default	» ?
Enable Wand	☑	Default	» ?
Enable WebSockets	☑	Default	» ?
Enable Webfonts	☑	Default	» ?
Enable config URL	☑	Default	» ?
Entities In Forms	☑	Default	» ?
Error Console Filter	mail=3	Default	» ?
Error Style File	:86)\Opera\styles\error.css' Choose...	Default	» ?
Extension Set Parent	Document Toolbar	Default	» ?
Fallback HTML Encoding	iso-8859-1	Default	» ?
Fast Forward Configuration	Ɔpera\Opera\fastforward.ini' Choose...	Default	» ?
First Update Delay	1000	Default	» ?
FlexRoot Max Width	0	Default	» ?
FlexRoot Min Width	0	Default	» ?
Force Encoding		Default	» ?
Format Unstyled XML	☑	Default	» ?
Geolocation site state	-1	Default	» ?

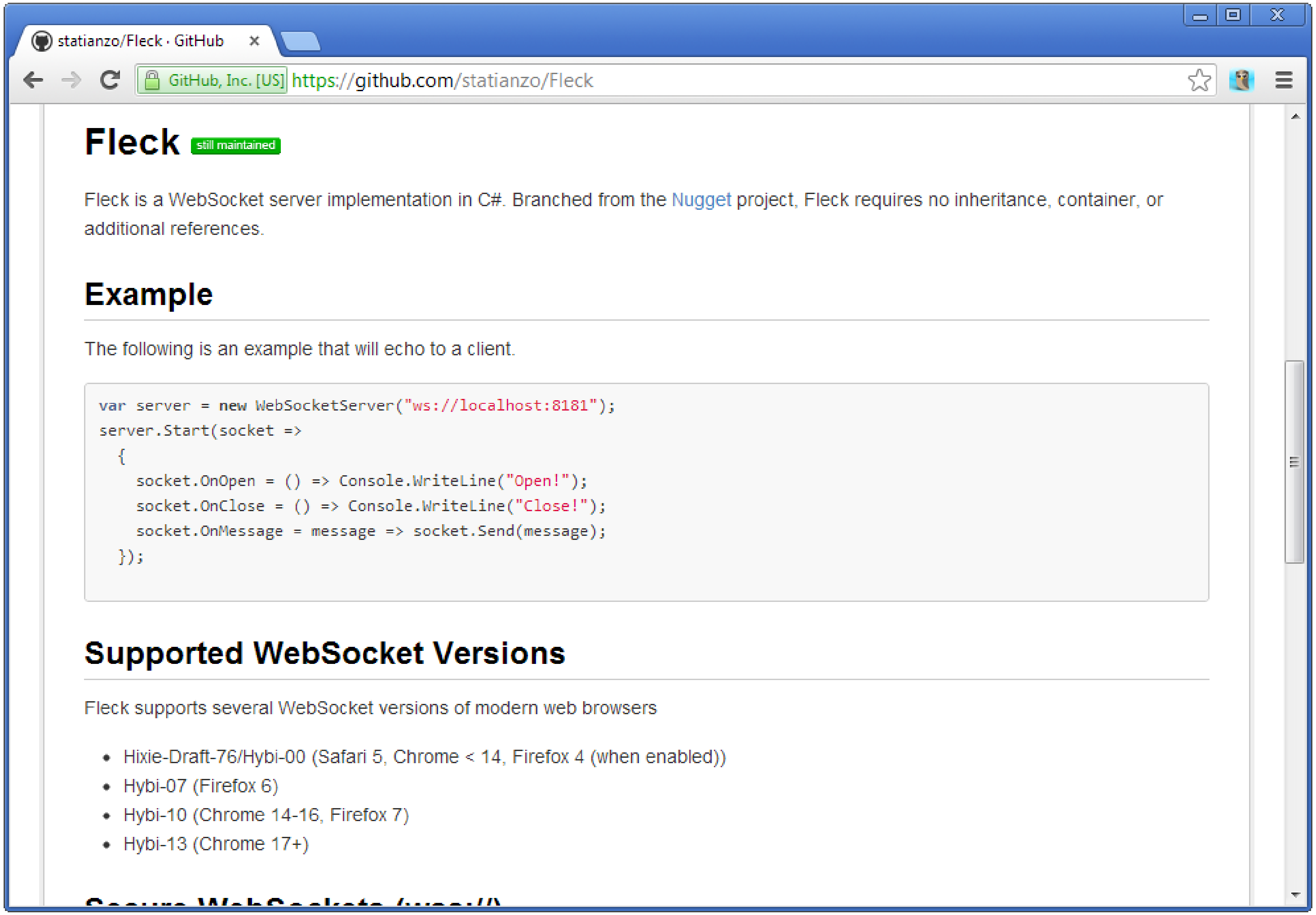
Enabling Web Sockets in Opera



config#UserPrefs|EnableWebSockets

Enable Trust Rating	☑	Default
Enable UI Animations	☑	Default
Enable Unite	☑	Default
Enable Usage Statistics	□	Default
Enable Wand	☑	Default
Enable WebSockets	☑	Default
Enable Webfonts	☑	Default
Enable config URL	☑	Default
Entities In Forms	☑	Default
Error Console Filter	mail=3	Default
Error Style File	:86)\Opera\styles\error.css' Choose...	Default
Extension Set Parent	Document Toolbar	Default
Fallback HTML Encoding	iso-8859-1	Default
Fast Forward Configuration	)pera\Opera\fastforward.ini' Choose...	Default
First Update Delay	1000	Default
FlexRoot Max Width	0	Default
FlexRoot Min Width	0	Default
Force Encoding		Default
Format Unstyled XML	☑	Default
Geolocation site state	-1	Default

Fleck



Fleck still maintained

Fleck is a WebSocket server implementation in C#. Branched from the Nugget project, Fleck requires no inheritance, container, or additional references.

Example

The following is an example that will echo to a client.

```
var server = new WebSocketServer("ws://localhost:8181");
server.Start(socket =>
  {
    socket.OnOpen = () => Console.WriteLine("Open!");
    socket.OnClose = () => Console.WriteLine("Close!");
    socket.OnMessage = message => socket.Send(message);
  });
```

Supported WebSocket Versions

Fleck supports several WebSocket versions of modern web browsers

- Hixie-Draft-76/Hybi-00 (Safari 5, Chrome < 14, Firefox 4 (when enabled))
- Hybi-07 (Firefox 6)
- Hybi-10 (Chrome 14-16, Firefox 7)
- Hybi-13 (Chrome 17+)

Demos

Summary

- Bi-directional full duplex client/server communication channel
- Much less overhead than HTTP
- Browser support is varied
- Simple API with powerful implications

For more in-depth online developer training visit

pluralsight

see what you can learn

on-demand content from authors you trust