

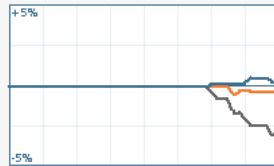
# Server push and web sockets

Real-time is instantly refreshing

KAAZING >K™

Hello Palo Alto | 3/3 | 1:10:34 PM | Google Talk Account | Password | About

Market Ticker



KZNG	16.57	-0.12	(-0.72%)
WSKT	12.74	-0.55	(-4.14%)
RTIM	15.70	0.05	(0.32%)

Currency Exchange

GBP / USD	13:10:34
SELL	BUY
1.62 <b>294</b>	50.0 <b>344</b>

EUR / USD	13:10:26
SELL	BUY
1.40 <b>250</b>	30.0 <b>280</b>



Keep the conversation going in real-time.

New York Times

The New York Times

**Life's Very Fine Lines**  
A song about the very fine lines in life.

**Swiss Voters Move to Restrict Pay of Executives**  
Ignoring a warning from the business lobby, Swiss residents voted to give shareholders a binding say on the overall pay packages for executives and directors.  
By RAPHAEL MINDER

**Who Should We Listen To?**  
A political appointment does not an expert make.  
By PAUL KRUGMAN

**37 Pakistanis Killed in Bombs in Shiite District of Karachi**  
Two powerful explosions ripped through a predominately Shiite neighborhood in Karachi on Sunday evening, in a bomb attack that killed at least 37 people and wounding at least 90, police and rescue officials said.  
By SALMAN MASOOD

**Condé Nast Invests in e-Commerce**  
Farfetch, an e-commerce site that serves independent boutiques, is getting a \$20 million investment.  
By SUZY MENKES

**A New Take on Tailoring**  
In Paris, a new take on tailoring is the story for winter 2013.  
By SUZY MENKES

Time-based Advertisement



Server Log

Logging...

Twitter

goings at &#8216;Downton Abbey' announced for the British drama's 4th season  
<http://t.co/1e1gZ31Fc>  
ABC News 6 Ways to Keep Airfare Costs Low  
<http://t.co/zxzjgBm4pY>  
msnbc Does \$791 billion buy a sense of safety? Check out how much the US has spent in the name of "homeland security."  
<http://t.co/YHb1RiD0dx>  
Los Angeles Times Texting while walking? Nevada assemblyman moves to ban it  
<http://t.co/X9mrhvVKim>

twitter

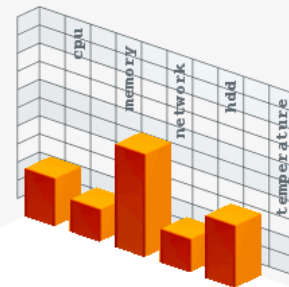
Time-based Advertisement

Make the Web a More Colorful Place



Paint it ORANGE. Make it KAAZING

Monitor



<http://kaazing.com>

# Responsive web apps, v1.0

- Problem: Client page needs info from server
- Solution: **AJAX** allows client to *pull* info
  - XMLHttpRequest makes asynchronous requests
    - Hacks to get around cross-domain restrictions
  - Uses standard HTTP request/response protocol
    - Small payload messages have high overhead
    - Latency introduced by HTTP processing



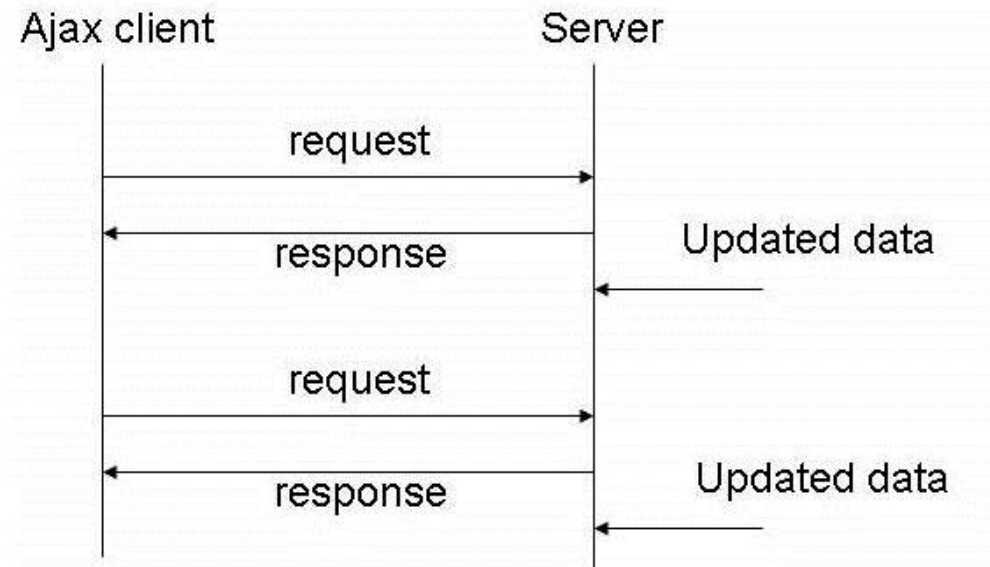
Children with

google will		<a href="#">Advanced Search</a>
google will eat itself	434,000 results	<a href="#">Preferences</a>
google will not load	10,500,000 results	<a href="#">Language Tools</a>
google will take over the world	16,000,000 results	
google will not open	81,000,000 results	by Google.
google will rule the world	12,500,000 results	
google will not search for chuck norris	286,000 results	
Advertising F	google will pay you to type	8,520,000 results <a href="#">Indonesia</a>
	google wills	2,190,000 results
	google will you marry me	429,000 results
	google will harm your computer	245,000 results
		<a href="#">close</a>



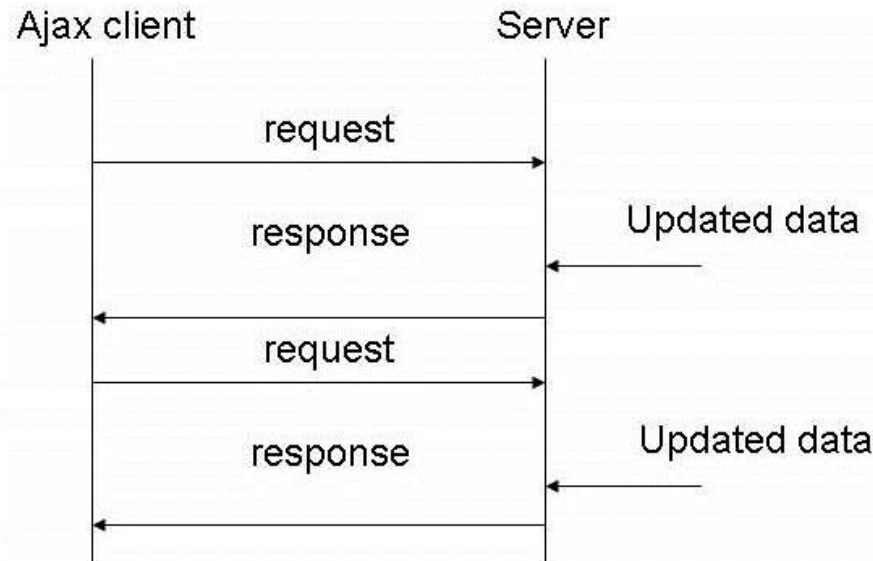
# Responsive web apps, v2.0

- **Problem:** Server needs to *push* info to client
  - e.g. update stock price, movement of players, etc.
- **Possible solutions:**
  - **Polling:** Client makes periodic AJAX requests
    - Works well if you know the correct polling interval
    - Otherwise wastes network/server resources



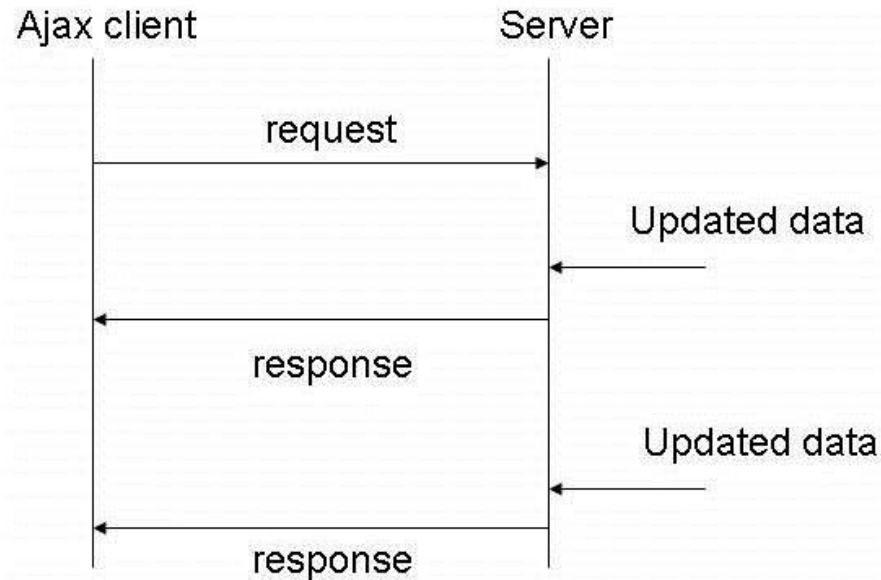
# Responsive web apps, v2.0

- **Problem:** Server needs to *push* info to client
  - e.g. update stock price, movement of players, etc.
- **Possible solutions:**
  - **Long-polling:** Client sends HTTP request, server waits until it has data to send in response
    - Hanging request may have high resource costs



# Responsive web apps, v2.0

- **Problem:** Server needs to *push* info to client
  - e.g. update stock price, movement of players, etc.
- **Possible solutions:**
  - **Streaming:** Server maintains open response continuously updated with push events
    - Subject to buffering by agents in network



# Streaming: HTTP response

- Response from server

- Status line:

- Protocol version, status code, status phrase

- Response headers: extra info

- Body: optional data

```
HTTP/1.1 200 OK
Date: Thu, 17 Nov 2011 15:54:10 GMT
Server: Apache/2.2.16 (Debian)
Last-Modified: Wed, 14 Sep 2011 17:04:27 GMT
Content-Length: 285

<html> ...
```

Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later

# Streaming: HTTP response

- Chunked response

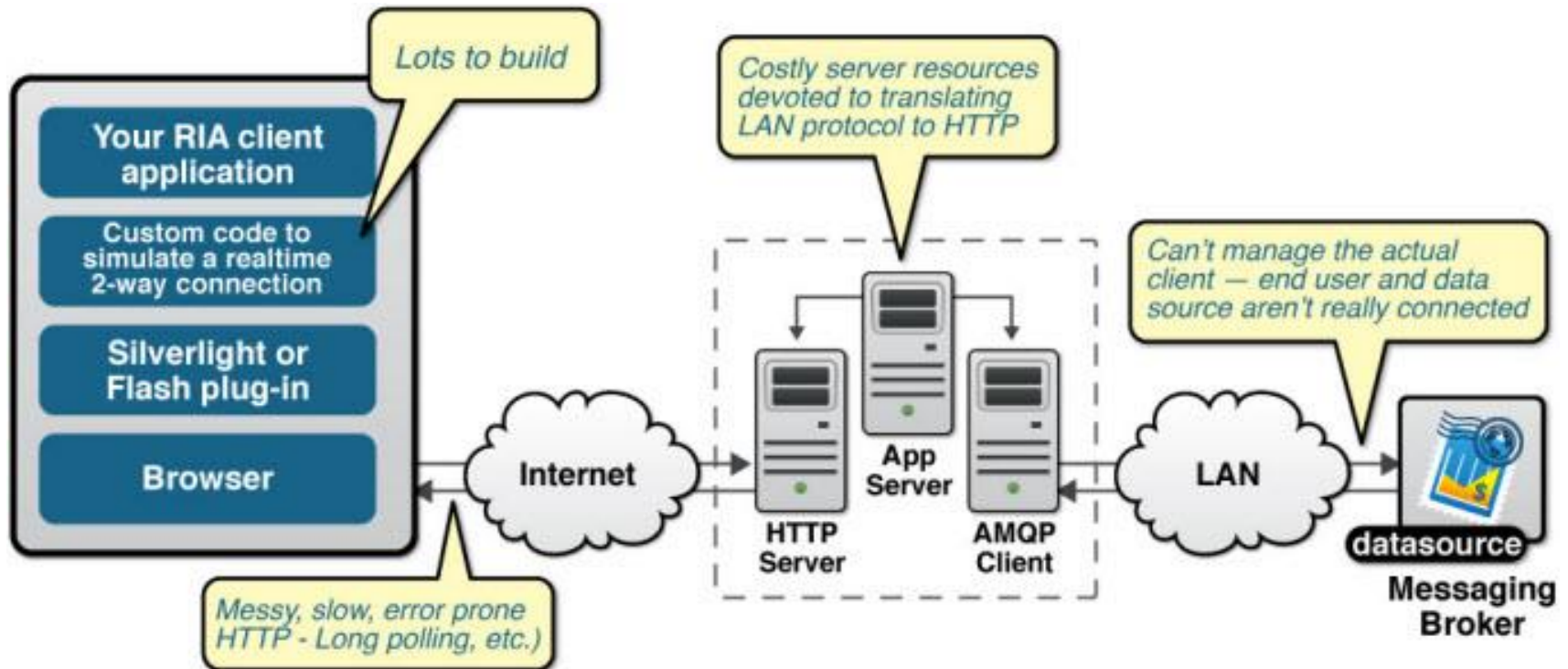
- Each chunk specifies size in hex, last chunk = 0

```
HTTP/1.1 200 OK
Date: Thu, 17 Nov 2011 15:54:10 GMT
Server: Apache/2.2.16 (Debian)
Last-Modified: Wed, 14 Sep 2011 17:04:27 GMT
Transfer-Encoding: chunked
29
<html><body><p>The file you requested is
5
3,400
23
bytes long and was last modified:
1d
Sat, 20 Mar 2004 21:12:00 GMT
13
.</p></body></html>
0
```

# Comet



- Comet (via polling or streaming)
  - Simulate bi-directional communication
    - Using HTTP request/response protocol
    - Often 2 connections: 1 downstream, 1 upstream
    - Resource expensive and error prone to write

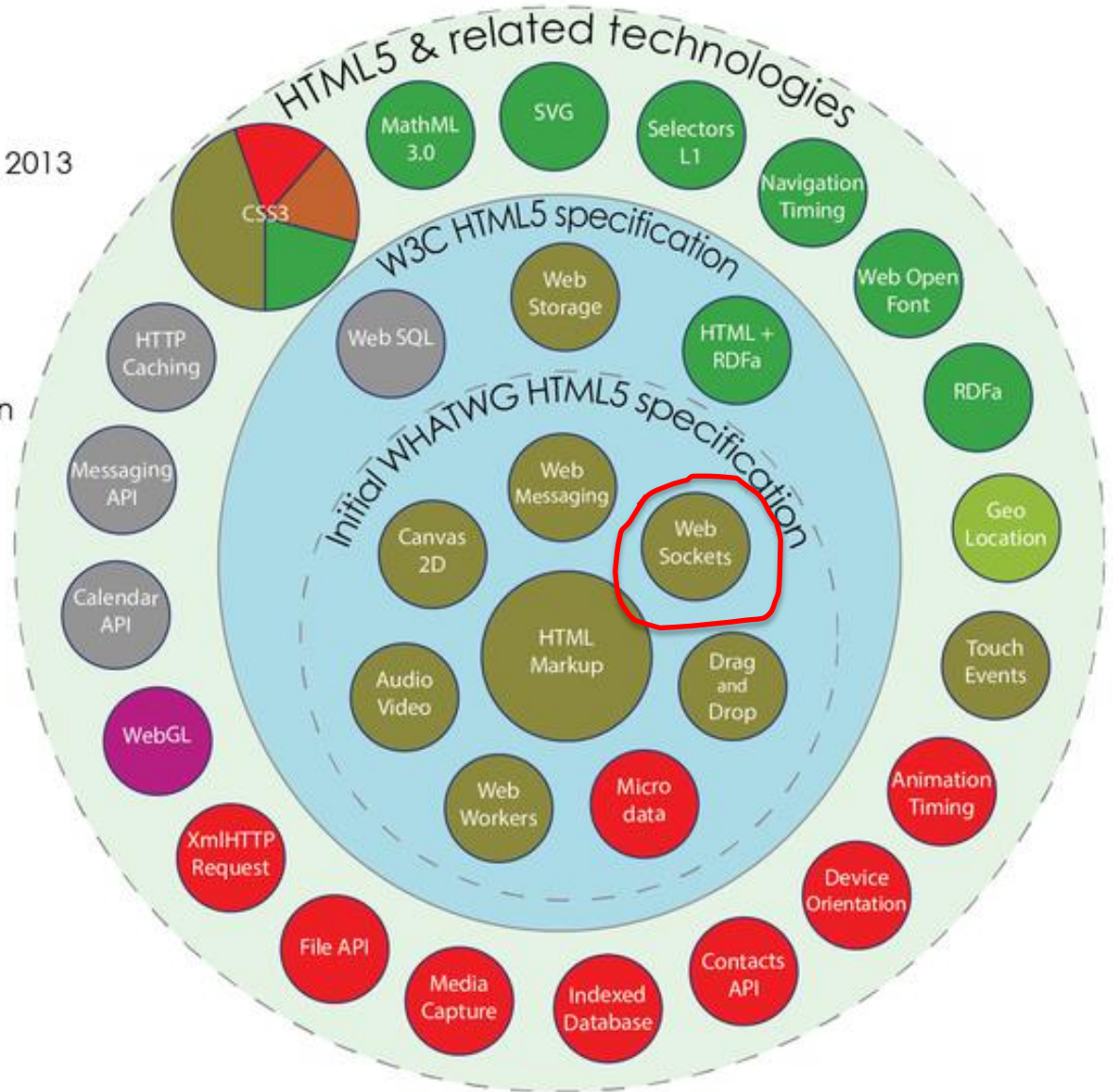




# HTML5

Taxonomy & Status on January 20, 2013

- W3C Recommendation
- Proposed Recommendation
- Candidate Recommendation
- Last Call
- Working Draft
- Non-W3C Specifications
- Deprecated



by Sergey Mavrody (cc) BY · SA

# Web Sockets - Candidate Recommendation

Global user stats\*:

Support:	72.03%
Partial support:	2.02%
Total:	74.05%

Bidirectional communication technology for web apps

Resources: [WebPlatform Docs](#) [Wikipedia](#) [Details on newer protocol](#) [WebSockets information](#)  
[has.js test](#)

	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser	Opera Mobile	Chrome for Android	Firefox for Android	IE Mobile
29 versions back			4.0										
28 versions back			5.0										
27 versions back		2.0	6.0										
26 versions back		3.0	7.0										
25 versions back		3.5	8.0										
24 versions back		3.6	9.0										
23 versions back		4.0	10.0										
22 versions back		5.0	11.0										
21 versions back		6.0 <sup>Moz</sup>	12.0										
20 versions back		7.0 <sup>Moz</sup>	13.0										
19 versions back		8.0 <sup>Moz</sup>	14.0										
18 versions back		9.0 <sup>Moz</sup>	15.0										
17 versions back		10.0 <sup>Moz</sup>	16.0										
16 versions back		11.0	17.0										
15 versions back		12.0	18.0		9.0								
14 versions back		13.0	19.0		9.5-9.6								
13 versions back		14.0	20.0		10.0-10.1								
12 versions back		15.0	21.0		10.5								
11 versions back		16.0	22.0		10.6								
10 versions back		17.0	23.0		11.0								
9 versions back		18.0	24.0		11.1								
8 versions back		19.0	25.0		11.5								
7 versions back		20.0	26.0	3.1	11.6			2.1					
6 versions back	5.5	21.0	27.0	3.2	12.0			2.2		10.0			
5 versions back	6.0	22.0	28.0	4.0	12.1	3.2		2.3		11.0			
4 versions back	7.0	23.0	29.0	5.0	15.0	4.0-4.1		3.0		11.1			
3 versions back	8.0	24.0	30.0	5.1	16.0	4.2-4.3		4.0		11.5			
2 versions back	9.0	25.0	31.0	6.0	17.0	5.0-5.1		4.1		12.0			
Previous version	10.0	26.0	32.0	6.1	18.0	6.0-6.1		4.2-4.3	7.0	12.1			
Current	11.0	27.0	33.0	7.0	19.0	7.0	5.0-7.0	4.4	10.0	16.0	33.0	26.0	10.0
Near future		28.0	34.0		20.0								
Farther future		29.0	35.0		21.0								
3 versions ahead		30.0	36.0										

# HTML5 Web Sockets

- **Web sockets:**
  - JavaScript interface for client-side
  - **Full-duplex** communication
    - Using a single object, send string or binary data
    - **Low latency, low header overhead** (strings = 2 bytes)
  - Initial **handshake over HTTP**
    - Upgraded to web socket protocol
      - Some proxies may not like and drop the connection
    - Runs on **port 80** allowing it to traverse NATs



*"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."*

*-Ian Hickson*

# Web socket protocol

- URL prefix:
  - ws:// for normal connections, wss:// for secure
- HTTP-compatible handshake:

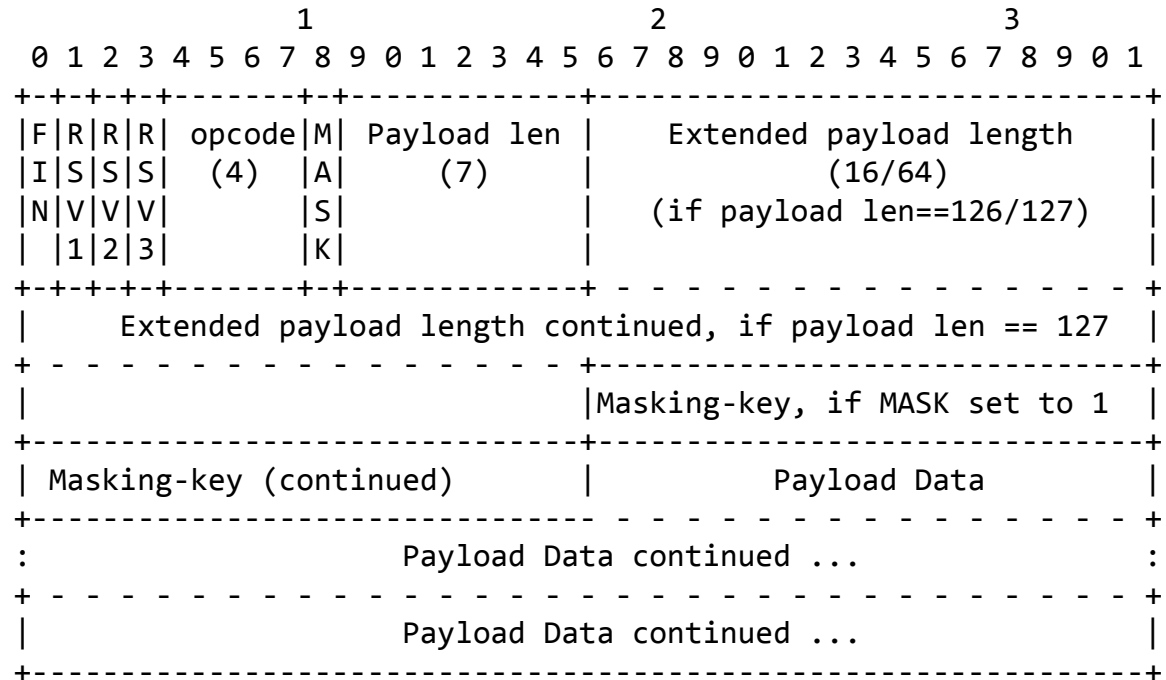
```
GET ws://echo.websocket.org/?encoding=text HTTP/1.1
Origin: http://websocket.org
Cookie: __utma=99as Connection: Upgrade
Host: echo.websocket.org
Sec-WebSocket-Key: uRovscZjNo1/umbTt5uKmw==
Upgrade: websocket
Sec-WebSocket-Version: 13
```

```
HTTP/1.1 101 WebSocket Protocol Handshake
Date: Fri, 10 Feb 2012 17:38:18 GMT
Connection: Upgrade
Server: Kaazing Gateway
Upgrade: WebSocket
Access-Control-Allow-Origin: http://websocket.org
Access-Control-Allow-Credentials: true
Sec-WebSocket-Accept: rLHCkw/SKs09GAH/ZSFhBATDKrU=
Access-Control-Allow-Headers: content-type
```

# Web socket protocol

- After handshake:
  - HTTP connection broken down
  - Replaced by WebSocket connection
    - Over the same TCP/IP connection
    - Update is one way, can't go back to HTTP

- Framing:



# Example messages

- A single-frame unmasked text message
  - 0x81 0x05 0x48 0x65 0x6c 0x6c 0x6f (contains "Hello")
- A fragmented unmasked text message
  - 0x01 0x03 0x48 0x65 0x6c (contains "Hel")
  - 0x80 0x02 0x6c 0x6f (contains "lo")
- Unmasked Ping request and masked Ping response
  - 0x89 0x05 0x48 0x65 0x6c 0x6c 0x6f (contains a body of "Hello")
  - 0x8a 0x85 0x37 0xfa 0x21 0x3d 0x7f 0x9f 0x4d 0x51 0x58 (contains a body of "Hello", matching the body of the ping)
- 256 bytes binary message in a single unmasked frame
  - 0x82 0x7E 0x0100 [256 bytes of binary data]
- 64KiB binary message in a single unmasked frame
  - 0x82 0x7F 0x0000000000010000 [65536 bytes of binary data]

# Web socket examples

## Echo Test

The first section of this page will let you do an HTML5 WebSocket test against the echo server. The second section walks you through creating a WebSocket application yourself.

You can also inspect WebSocket messages using your browser.

## Try it out



Location:  
ws://echo.websocket.org  
 Use secure WebSocket (TLS)  
Connect Disconnect

Message:  
Rock it with HTML5 WebSocket  
Send

Log:  
CONNECTED  
SENT: Rock it with HTML5 WebSocket  
RESPONSE: Rock it with HTML5 WebSocket

Clear log

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

Real-time is instantly refreshing

Market Ticker

KZING	16.57	-0.12 (-0.72%)
WSKT	12.74	-0.55 (-4.14%)
RTBH	15.70	0.05 (0.32%)

Life's Very Fine Lines  
A song about the very fine lines in life.

Swiss Voters Have to Restrict Pay of Executives  
Ignoring a warning from the business lobby, Swiss residents voted to give shareholders a binding say on the overall pay packages for executives and directors.

Who Should We Listen To?  
A political appointment does not an expert make.

37 Pakistanis Killed in Blasts in Shiite District of Karachi  
Two powerful explosions ripped through a predominately Shiite neighborhood in Karachi on Sunday evening, in a bomb attack that killed at least 37 people and wounding at least 90, police and rescue officials said.

Condé Nast Invests in e-Commerce  
Farfetch, an e-commerce site that serves independent boutiques, is getting a \$20 million investment.

A New Take on Tailoring  
In Paris, a new take on tailoring is the story for winter 2013.

Time-based Advertisement  
ARCHITECTURE · MOBILE · WEB APPS · COMMUNICATION · BROWSER SUPPORT · BUSINESS IMPACT

Server Log  
Logging...

Keep the conversation going in real-time.

<http://demo.kaazing.com/livefeed/>

RUMPETROLL

Instructions  
Click & hold to swim - type to chat  
Type 'name: yourname'  
@names are reserved for Twitter users. Sign in.

Guest 356620

lesperm faic

Spermy

<http://rumpetroll.com/>

<http://labs.dinahmoe.com/plink/>

<http://www.youtube.com/watch?v=64TcBiqmVko>

<http://www.html5rocks.com/en/tutorials/websockets/basics/>

# WebSocket interface

```
[Constructor(DOMString url, optional (DOMString or DOMString[]) protocols)]
interface WebSocket : EventTarget {
  readonly attribute DOMString url;

  // ready state
  const unsigned short CONNECTING = 0;
  const unsigned short OPEN = 1;
  const unsigned short CLOSING = 2;
  const unsigned short CLOSED = 3;
  readonly attribute unsigned short readyState;
  readonly attribute unsigned long bufferedAmount;

  // networking
      attribute EventHandler onopen;
      attribute EventHandler onerror;
      attribute EventHandler onclose;
  readonly attribute DOMString extensions;
  readonly attribute DOMString protocol;
  void close([Clamp] optional unsigned short code, optional DOMString reason);

  // messaging
      attribute EventHandler onmessage;
      attribute DOMString binaryType;
  void send(DOMString data);
  void send(Blob data);
  void send(ArrayBuffer data);
  void send(ArrayBufferView data);
};
```



# Simple text echo client

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<script>
function init()
{
  websocket          = new WebSocket("ws://echo.websocket.org/");
  websocket.onopen   = function(e) { onOpen(e)    };
  websocket.onclose  = function(e) { onClose(e)   };
  websocket.onmessage = function(e) { onMessage(e) };
  websocket.onerror  = function(e) { onError(e)   };
}

function onOpen(e)
{
  writeToScreen("CONNECTED");
  message = "Hello world!";
  writeToScreen("SENT: " + message);
  websocket.send(message);
}

function onClose(e)
{
  writeToScreen("DISCONNECTED");
}

function onMessage(e)
{
  writeToScreen('RESPONSE: ' + e.data);
  websocket.close();
}

function onError(e)
{
  writeToScreen('ERROR: ' + e.data);
}

function writeToScreen(message)
{
  document.getElementById("output").innerHTML +=
    message + "<br />";
}
window.addEventListener("load", init, false);
</script>
</head>

<body>
<h2>WebSocket Test</h2>
<div id="output"></div>
</body>
</html>
```

# Supported data types

- In latest spec, send data as:

- Text
- ArrayBuffer
- Blob

```
// Sending String
connection.send('your message');

// Sending canvas ImageData as ArrayBuffer
var img = canvas_context.getImageData(0, 0, 400, 320);
var binary = new Uint8Array(img.data.length);
for (var i = 0; i < img.data.length; i++)
{
    binary[i] = img.data[i];
}
connection.send(binary.buffer);

// Sending file as Blob
var file = document.querySelector('input[type="file"]').files[0];
connection.send(file);
```

```
// Setting binaryType to accept received binary as either 'blob' or 'arraybuffer'
connection.binaryType = 'arraybuffer';
connection.onmessage = function(e)
{
    console.log(e.data.byteLength); // ArrayBuffer object if binary
};
```

<http://www.html5rocks.com/en/tutorials/websockets/basics/>

# Client programming in practice

- **WebSockets relatively new**
  - Browser support is **widespread but not ubiquitous**
  - Standard not yet finalized
  - **Network proxies may mess things up**
- **Other techniques more mature**
  - XMLHttpRequest
- **Possible option: use a 3<sup>rd</sup> party client library**
  - Library provides semantics of bi-directional communication, but encapsulates details
  - e.g. socketio, cometd, Hookbox, orbited

# Web socket server

- The server side
  - You need **server-side support!**
    - Must support a large number of open WebSocket Connections
    - Traditional stacks (e.g. LAMP) do not deal well with this
- Apache options:
  - apache-websocket
    - Apache module for WebSocket protocol
    - Develop your own module (in C) for app-specific details
  - pywebsocket
    - As an Apache module or as a standalone server
    - Requires mod\_python

# Other server options...

- C/C++
  - [libwebsockets](#)
  - [Mongoose](#)
  - [POCO C++ Libraries](#)
  - [Tufão](#)
  - [Wslay](#)
  - [QtWebsocket](#)
- Erlang
- Go
  - [Yaws](#)
- Go
  - [go.net/websocket](#)
  - [webrocket](#)
- Haskell
  - [websockets](#)
- Java
  - [Apache Tomcat 7](#)
  - [Play Framework](#)
  - [Atmosphere](#)
  - [Bristleback](#)
  - [GlassFish 3.1](#), [Grizzly](#)
  - [HLL WebSockets](#)
  - [JBoss 7](#)
  - [Jetty 7](#)
  - [jWebsocket](#)
  - [Netty 3.3](#)
  - [MigratoryData WebSocket Server](#)
- .NET Framework
  - [Internet Information Services \(IIS\) 8](#), [ASP.NET 4.5](#)
  - [Windows Communication Foundation 4.5](#) through [NetHttpBinding](#)
  - [Fleck](#)
  - [SuperWebSocket](#)
  - [XSockets.NET](#)
- Clojure
  - [http-kit](#)
  - [aleph](#)
- Nginx
  - [Proxy \(since version 1.3.13\)](#)
  - [Push Stream \(3-rd party module\)](#)
- Node.js
  - [Socket.IO](#)
  - [WebSocket-Node](#)
- Objective-C
  - [SocketRocket](#)
  - [BLWebSocketsServer](#)
- Perl
  - [Mojolicious](#)
  - [PocketIO](#)
- PHP
  - [php-websocket](#)
  - [Ratchet](#)
- Python
  - [WebSocket-for-Python](#)
  - [txWS](#)
  - [AutobahnPython](#)
- Ruby
  - [EM-WebSocket](#)
- Other
  - [apache-websocket](#)
  - [mod\\_websocket](#) for [lighttpd](#)
  - [nginx supports websocket since version 1.3](#)

# Summary

- Responsive interactive web apps
  - Requires low latency bi-directional communication
  - Existing solutions:
    - Ajax polling, long polling, streaming
    - But these are really hacks working within an ill-suited HTTP request/response framework
  - HTML5 web sockets:
    - Simple client -side API
    - Requires server that supports web sockets
    - You have to develop app-specific logic in some way
      - e.g. Apache module, Java servlet, ...