Using WebSockets with ColdFusion

Raymond Camden | Developer Evangelist

Twitter: @cfjedimaster

Who am I?

- Raymond Camden
- Developer Evangelist for Adobe
- www.raymondcamden.com
- @cfjedimaster

What the heck is a Web Socket?

"WebSocket is a web technology providing for multiplexing bi-directional, full-duplex communications channels over a single TCP connection. "

-Wikipedia

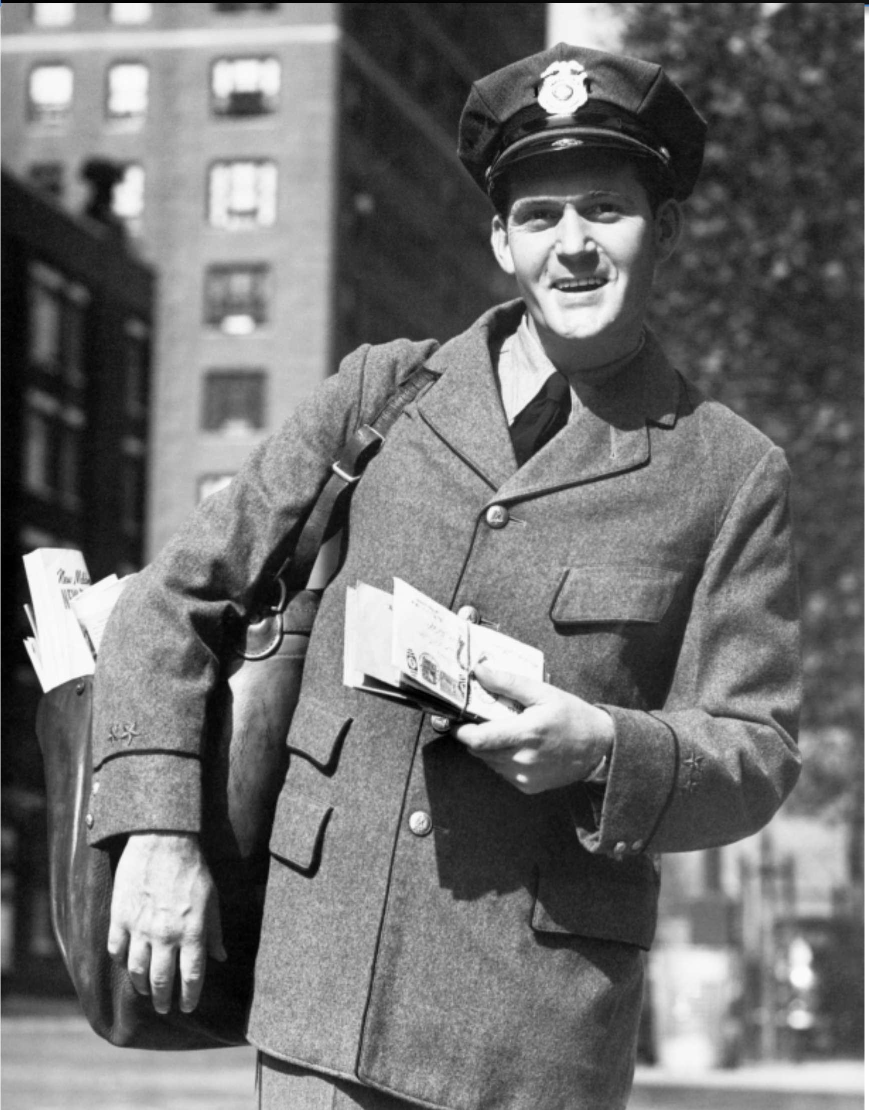
No, seriously, what is it?

WebSockets are a way to create JavaScript applications that have a true, open, connection to the server. This means they can receive updates instantly and broadcast to other clients.

Ok, dude, what?

```
setInterval(checkForStuff, 2000);
function checkForStuff() {
 someAjaxCallToTheServer();
}
```

Remember this guy?

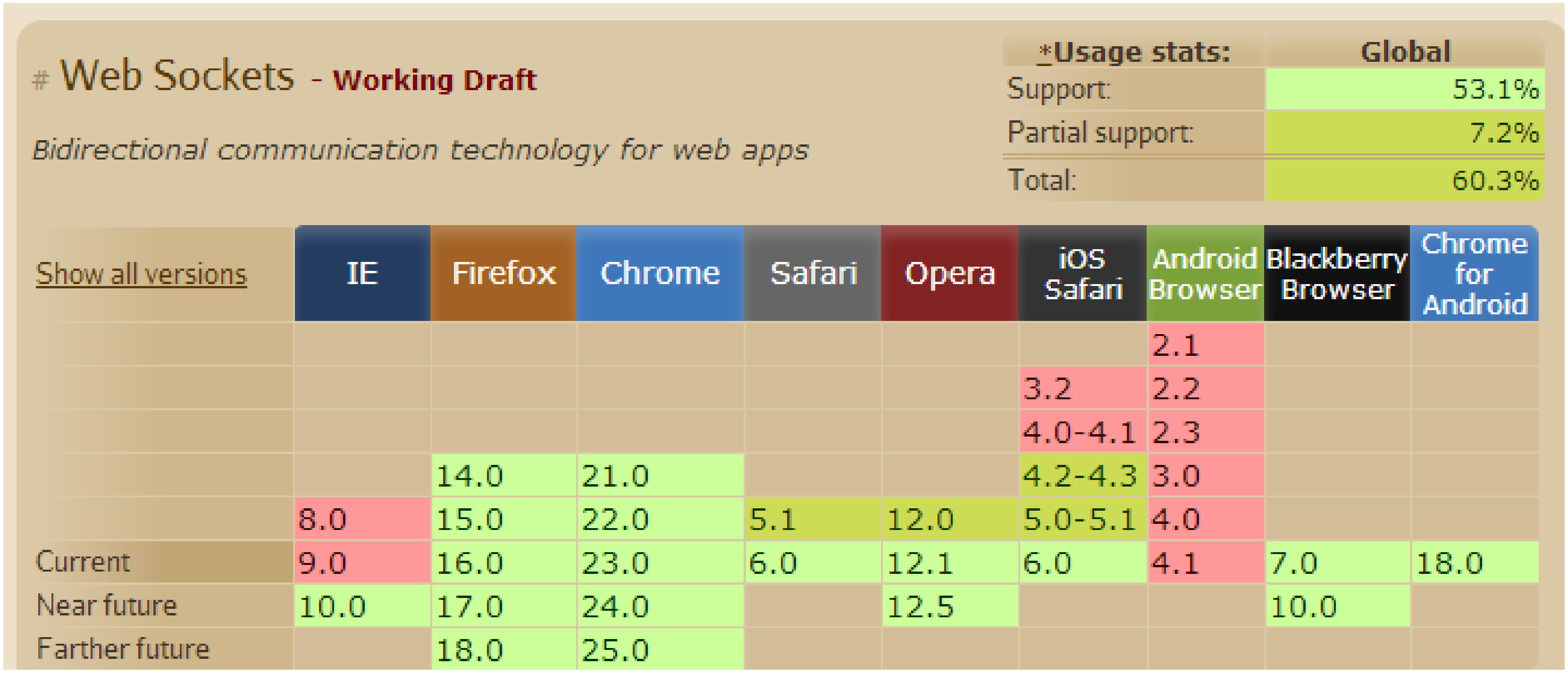


The New Hotness

- Allow me to open a connection
- I'm told when new stuff is broadcast (server or other clients)
- I can broadcast my own stuff
- All kinds of fun filtering/organizing possible (think switchboard)

Communication via Phone

Is your browser cool?



Web Sockets - Working Draft

Bidirectional communication technology for web apps

*Usage stats:	Global
Support:	53.1%
Partial support:	7.2%
Total:	60.3%

Show all versions	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Android Browser	Blackberry Browser	Chrome for Android
							2.1		
						3.2	2.2		
						4.0-4.1	2.3		
		14.0	21.0			4.2-4.3	3.0		
	8.0	15.0	22.0	5.1	12.0	5.0-5.1	4.0		
Current	9.0	16.0	23.0	6.0	12.1	6.0	4.1	7.0	18.0
Near future	10.0	17.0	24.0		12.5			10.0	
Farther future		18.0	25.0						

!websocket

- Fallback to Flash
- 100% of your code still works
- Support for showing a message to the poor saps left out…

The Details

- Client side and server side
- ColdFusion tags and JavaScript
- You will be writing JavaScript

Step One

- Application.cfc defines valid channels
- Channels are the most broad, most high level organization for WebSockets
- Defined as an array of structs

Example

```
this.wschannels = [{name="Sports"},
                   {name="Stocks"},
                   {name="Weather"}];
```

Step Two

- Your CFM defines a websocket via the new <cfwebsocket> tag
- Give it a name that sets up the JavaScript handle
- Tell it what to run when a message comes in
- Tell it what channel to connect to
- (There's more options)

Example

```
<cfwebsocket name="myWS"
onMessage="messageHandler"
subscribeTo="news">
```

Demo

- /example1

Using the JavaScript API

- publish - send a message (anything!)
- openConnection/closeConnection - pick up or hang up the phone
- subscribe/unsubscribe - connect (or disconnect) from a channel
- getSubcriptions - what I'm subscribed to
- getSubscriberCount - how many people are listening

More...

- authenticate - used for secured channels
- invoke and invokeAndPublish - used to communicate to a CFC

Demo

- /example2

CFC Handlers

- Give you server-side control over your websockets
- Must extend `CFIDE.websocket.ChannelListener`
- Some map to JavaScript functions
- Define the use of a handler in your Application.cfc
- Cached!!

Methods

- allowSubscribe - can I join the party?
- allowPublish - can I say something?
- beforePublish - format the message
- canSendMessage - can I hear something?
- beforeSendMessage - client specific formatting/modification
- afterUnsubscribe

Demo

- /example3

Server-Side Functions

- wsGetSubscribers(channel)
- wsPublish(channel, msg)
- wsGetAllChannels

Demo

- /example4

Filtering Options

- Multiple Channels
  - News, Weather, Sports, and Beer
- Manual processing
  - Messages can include custom data
- Subchannels
- Selectors

Subchannels

- You must subscribe to a channel defined in App.cfc, ala "news"
- But you can subscribe to a "dot path" under this: "news.sports"
- And go as far as you want: "news.sports.american.football"

Subchannels (2)

- You get messages for your subscription and "lower" nodes.
- Subscribed to news and you get news, news.sports, news.tech
- Subscribed to news.sports, you won't get news or news.tech
- Subscribed to news.sports, you will get news.sports.football

Demo

- /example5

Selectors

- Allow for more precise targeting
- Applies to publishing/receiving
- Selector is a basic conditional
- property <some comparison> value
- ColdFusion expressions, not JavaScript (no < or >)

Selector Example

- Ch: Products, Selector: price lt 100
- Ch: Stocks, Selector: change gt 10
- Ch: Scores, Selector: sport eq 'football'

Demo

- /example6

Security

Authentication

- Via onWSAuthenticate and JavaScript code (in other words, login with your fancy WebSocket app)
- Via an existing login, but cflogin only

onWSAuthenticate

- New method of Application.cfc
- Passed username, password, connectionInfo
  - connectionInfo is a struct you modify
  - must set .authenticated=true at minimum
- In JavaScript, use authenticate method
- Note! CFC handler MUST check in allowSubscribe

Demo

- /example7

SingleSignOn Mode

- Basically, this mode works with an *existing* authenticated user
- Must work along with cflogin-based security

Demo

- /example8

Security

- Remember - your WebSocket JavaScript variable is manipulable
- Show the console hack in chat
- This is all as secure as any other Ajax application

Evil User Demo!

invokeAndPublish

- Used to run an adhoc CFC server to generate a message on a channel
- Usage:
  `myWS.invokeAndPublish("channel", "cfcname", "method", arrayOfArgs, structOfCustomHeaders)`
- Also runs `allowPublish()` in your CFC handler
- Cached

Demo

- /example9

Point2Point WebSocket

- Just you and the server (oh, how sweet)
- No channels involved - just a server CFC
- CFC can return messages, and make new ones via wsSendMessage

Demo

- /example10

Fallback

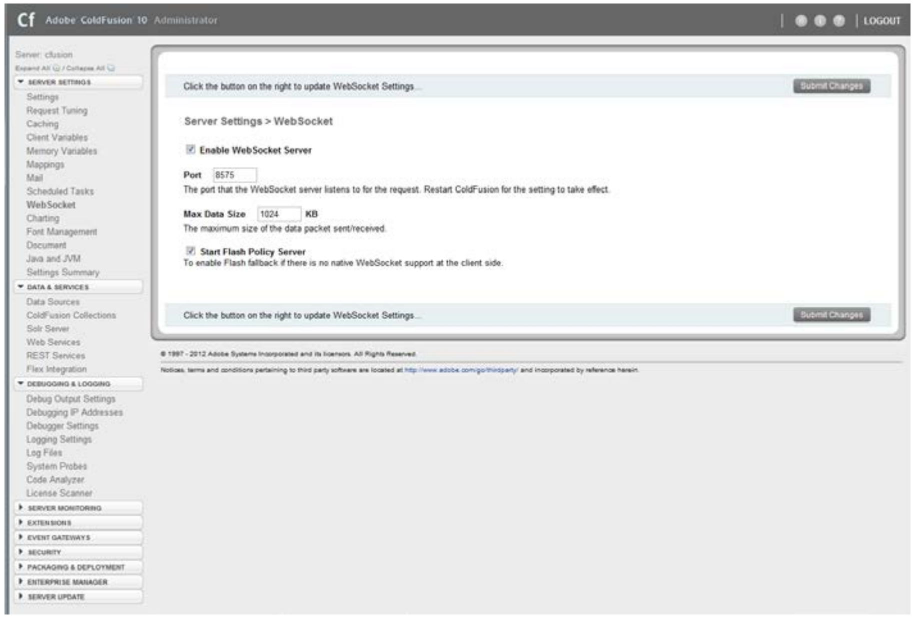
- If no websockets…
- Use Flash…
- If no Flash…
- messageHandler gets something
- or you can use onError

Demo

- /example11

CF Administrator Options

- Global enable/disable
- Set port and max data size
- Enable Flash fallback



Click the button on the right to update WebSocket Settings...

Server Settings > WebSocket

[✓] Enable WebSocket Server

Port 8575
The port that the WebSocket server listens to for the request. Restart ColdFusion for the setting to take effect.

Max Data Size 1024 KB
The maximum size of the data packet sent/received.

[✓] Start Flash Policy Server
To enable Flash fallback if there is no native WebSocket support at the client side.

Click the button on the right to update WebSocket Settings...

© 1997 - 2012 Adobe Systems Incorporated and its licensors. All Rights Reserved.

Notices, terms and conditions pertaining to third party software are located at http://www.adobe.com/go/thirdparty/ and incorporated by reference herein.

Chrome Dev Tools

- Just show it…

Any questions?