About me
Web application developers
HTML 5!
improved JavaScript implementations!
WebSockets!
WebSockets?
WebSockets

a technology that enables bidirectional communication between web browsers and server-side processes
“TCP for the web”
Push technology
Client-server communication

AJAX

Comet

WebSockets
Example
Multi-player game
Rock Paper Scissors Lizard Spock
Specifications

- WebSockets API
- WebSockets protocol
```javascript
var sock = new WebSocket('wss://www.oscon.com/chat');

sock.onopen = function(evt) { alert('open!'); };

sock.onmessage = function(evt) {
    alert('Message: ' + evt.data);
}

sock.onerror = function(evt) { alert('error!'); };

sock.onclose = function(evt) { alert('closed!'); }

sock.send("Hello OSCON!");
sock.close();
```
WebSocket URLs

- ws://example.com/demo
- wss://example.com/demo
event handlers

• `onopen`
• `onmessage`
• `onerror`
• `onclose`
Specifications

• WebSockets API
• WebSockets protocol
Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 22, 2011

I. Hickson
Google, Inc.
July 21, 2010

The WebSocket protocol
draft-ietf-hybi-thewebsocketprotocol-latest

Abstract

The WebSocket protocol enables two-way communication between a user agent running untrusted code running in a controlled environment to a remote host that has opted-in to communications from that code. The security model used for this is the Origin-based security model commonly used by Web browsers. The protocol consists of an initial handshake followed by basic message framing, layered over TCP. The goal of this technology is to provide a mechanism for browser-based applications that need two-way communication with servers that does not rely on opening multiple HTTP connections (e.g. using XMLHttpRequest or <iframe>s and long polling).
WebSockets protocol

- ports 80 and 443
- HTTP “upgrade” handshake
Browser request

GET /test HTTP/1.1
Upgrade: WebSocket
Connection: Upgrade
Origin: http://www.oscon.com/chat
Host: www.oscon.com
Content-Length: 0
HTTP/1.1 101 Web Socket Protocol Handshake
Upgrade: WebSocket
Connection: Upgrade
Server: FooServer/1.5
WebSocket-Location: ws://www.oscon.com/chat
WebSocket-Origin: http://www.oscon.com/chat
Content-Length: 0
Date: Fri, 08 May 2010 07:23:58 GMT
the WebSocket network protocol is still changing!
Protocol evolution

- draft-hixie-thewebsocketprotocol-75
- draft-hixie-thewebsocketprotocol-76
- draft-ietf-hybi-thewebsocketprotocol-00
Developers should be aware that starting from WebKit nightly build r59903 and Chrome 6.0.414.0 (r47952), the client will talk to a server using -76 version of the protocol, so it will fail to open WebSocket connections with a WebSocket server based on draft-hixie-thewebsocketprotocol-75. Since -75 version of the protocol is obsoleted and no longer supported in future version of browsers, to support new clients you need to update the server implementation. (Note that Chrome 5 uses -75 version of protocol)

The WebSocket protocol is still actively being changed. Until there is more consensus, we will continue to update our implementation to follow the latest draft of specification, rather than worrying about breaking changes.

Browser support

- Google Chrome 4.0.249.0 and higher
- Safari 5.0
- Firefox: 4.0 beta 1
- Internet Explorer 9: TBD
Apple decided to disable WebSockets in iOS 4. What's strange was that WebSocket support was in all the previous dev releases - oh well...
WebSockets on the server-side
As a result of a security breach on 4th April 2010, the Apache Infrastructure Team recommends that all Bugzilla users change their passwords as a precautionary measure. Please see the Infrastructure Blog for further information.

Bug List: (This bug is not in your last search results)  Show last search results

Bug 47485 - HTML5 Websocket implementation

Status: NEW

Product: Apache httpd-2
Component: All
Version: 2.3-HEAD
Platform: All

Importance: P3 enhancement (vote)
Target Milestone: ---
Assigned To: Apache HTTPD Bugs Mailing List

URL: http://tools.ietf.org/html/draft-hixi...
Keywords:

Depends on: 
Blocks: 

Last modified: 2010-06-30 11:39:15 EDT

Reported: 2009-07-06 17:54 EDT by Senthilkumar
Modified: 2010-06-30 11:39 EDT (History)
CC List: 9 users (show)
pywebsocket

What is it?

The pywebsocket project aims to provide a Web Socket extension for Apache HTTP Server, mod_pywebsocket.

mod_pywebsocket is intended for testing or experimental purposes. mod_python is required. For wss, mod_ssl is also required.

How can I use it?

To try mod_pywebsocket, please do:

```
svn checkout http://pywebsocket.googlecode.com/svn/trunk/ pywebsocket-read-only
```

and follow the instructions in pywebsocket-read-only/src/README.

To run mod_pywebsocket as an Apache HTTP Server extension module, please read comments in pywebsocket-read-only/mod_pywebsocket/__init__.py.

To run mod_pywebsocket as a standalone server (i.e., not using Apache HTTP Server), please...
django-websocket 0.3.0

Websocket support for django.

The `django-websocket` module provides an implementation of the WebSocket Protocol for django. It handles all the low-level details like establishing the connection through sending handshake reply, parsing messages from the browser etc...

It integrates well into django since it provides easy hooks to receive WebSocket requests either for single views through decorators or for the whole site through a custom middleware.

Usage

You can use the `accept_websocket` decorator if you want to handle websocket connections just for a single view - it will route standard HTTP requests to the view as well. Use `require_websocket` to only allow WebSocket connections but reject normal HTTP requests.

You can use a middleware if you want to have WebSockets available for all URLs in your application. Add `django_websocket.middleware.WebSocketMiddleware` to your `MIDDLEWARE_CLASSES` setting. This will still reject websockets for normal views. You have to set the `accept_websocket` attribute on a view to allow websockets.
WebSockets are one of the most underappreciated innovations in HTML5. Unlike local storage, canvas, web workers, or even video playback, the benefits of the WebSocket API are not immediately apparent to the end user. In fact, over the course of the past decade we have invented a dozen technologies to solve the problem of asynchronous and bi-directional communication between the browser and the server: AJAX, Comet & HTTP Streaming, BOSH, ReverseHTTP, WebHooks & PubSubHubbub, and Flash sockets amongst many others. Having said that, it does not take much experience with any of the above to realize that each has a weak spot and none solve the fundamental problem: web-browsers of yesterday were not designed for bi-directional communication.
Use custom exceptions rather than RuntimeError

mloughran (author)
Tuesday Nov 24, 2009

Jetty WebSocket Server

Jetty-7.0.1 has been extended with a WebSocket server implementation based on the same scalable asynchronous IO infrastructure of Jetty and integrated into the Jetty Servlet container.

WebSocket came out of work on HTML5 by the What Working Group to specify a mechanism to allow two way communications to a browsers. It is currently being standardized at the W3C for the WebSocket API and by the IETF for the WebSocket protocol and is soon to be supported by releases of Firefox, and Chromium. While I have significant concerns about the websockets protocol, it is important that server concerns are considered in the standardization process. Thus to follow the IETF model of "rough consensus and working code", it is important that Jetty has a working implementation of the protocol.

The key feature of the Jetty WebSocket implementation is that it is not another separate server. Instead it is fully integrated into the Jetty HTTP server and servlet container. so a Servlet or Handler can process and accept a request to upgrade a HTTP connection to a WebSocket connection. Applications components created by standard web applications can then send and receive datagrams over the WebSocket with non blocking sends and receives.

Below is an example of a SIMPLE "Chat" application written using the Jetty WebSocketServlet, which can handle normal doGet style requests, but will call doWebSocketConnect if an upgrade request is received:
# Package org.eclipse.jetty.websocket

## Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSocket</td>
</tr>
<tr>
<td>WebSocket.Outbound</td>
</tr>
<tr>
<td>WebSocketParser.EventHandler</td>
</tr>
</tbody>
</table>

## Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSocketBuffers</td>
<td>The WebSocket Buffer Pool.</td>
</tr>
<tr>
<td>WebSocketConnection</td>
<td></td>
</tr>
<tr>
<td>WebSocketFactory</td>
<td>Factory to create WebSocket connections</td>
</tr>
<tr>
<td>WebSocketGenerator</td>
<td>WebSocketGenerator.</td>
</tr>
<tr>
<td>WebSocketHandler</td>
<td></td>
</tr>
<tr>
<td>WebSocketParser</td>
<td>Parser the WebSocket protocol.</td>
</tr>
<tr>
<td>WebSocketServlet</td>
<td>Servlet to upgrade connections to WebSocket</td>
</tr>
</tbody>
</table>
No standard server-side Java API
Other Java projects

• Glassfish / Grizzly
• jWebSocket
• JBoss Netty
• Caucho Resin
Conclusion

• WebSockets is an emerging technology
• JavaScript API is easy to learn
• JavaScript API is easy to use
• Lots of choices on the server-side
Thank you

sean@seansullivan.com
Bonus slides!
Programming WebSockets

Sean Sullivan (-)
10:40am Thursday, 07/22/2010
JavaScript
Location: Portland 255

WebSockets is an exciting new technology that enables bidirectional communication between web applications and server-side processes. Google’s Chrome browser already provides WebSockets and developers can expect to see the technology in other browsers in 2010. This presentation will cover the WebSocket protocol, JavaScript API, and server-side implementations.
jQuery

- http://code.google.com/p/jquery-graceful-websocket/
- http://code.google.com/p/jquery-websocket/
Changing the size of Flash to 8x8 so that ClickToFlash detects it as invisible Flash.

gimite (author)
April 08, 2010

Initial version. Checked only with Chrome. [gimite]
Push technologies

• Flash sockets
• Silverlight duplex services
• Comet
• WebSockets
WebSockets and HTML5

“At last week's telecon, while discussing ISSUE-64, it was proposed that we declare WebSocket (both API and protocol) out of scope for HTML5. Since the API and protocol have been in separate specs for some time, this would have no immediate material effect. However, it would prevent us from putting WebSocket back in the main HTML5 spec in the future, unless new information came to light which would allow us to reopen the decision.”

September 9 2009

http://www.w3.org/html/wg/tracker/issues/64
WebSockets and HTML5

“Since there was no objection, the resolution has now passed.”

September 23 2009

http://www.w3.org/html/wg/tracker/issues/64
Security

- same-origin policy applies
- use `wss://` if you want a secure connection
Demo:

HTML5 + WebSockets
Quake II and the Quake logo are trademarks of id Software.

Note: We recently reset the repository to clean some things up, so if you cloned it you may find that the commit ids no longer match, and you'll need to re-clone. Our apologies for any inconvenience.

Quake II GWT Port

The Quake II GWT port brings the 3d gaming experience of Quake II to the browser.

In the port, we use WebGL, the Canvas API, HTML 5 <audio> elements, the local storage API, and WebSockets to demonstrate the possibilities of pure web applications in modern browsers such as Safari and Chrome.

The port is based on the Jake2 project, compiled to Javascript using the Google Web Toolkit (GWT). Jake 2 is a Java port of the original Quake II source code, which was open sourced by id software.

To make the Jake 2 code work with GWT, we have:

- Created a new WebGL based renderer
- Ported the network layer for multiplayer games from UDP to the WebSocket API
- Made all resource loading calls asynchronous
- Created a GWT implementation of Java nio buffers based on WebGL arrays (to be ported to ECMAScript Typed Arrays)