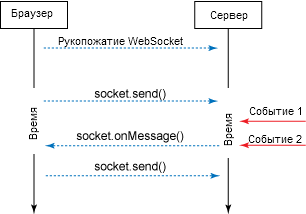
Web sockets

Метод WebSockets, который появился в HTML5 - это гораздо более молодой метод Reverse Ajax, чем Comet. WebSockets создает двунаправленные, дуплексные каналы связи, и многие браузеры (Firefox, Google Chrome и Safari) уже поддерживают его. Соединение открывается посредством HTTP-запроса со специальными заголовками, который называется рукопожатием WebSockets. Это соединение сохраняется постоянно, и через него можно записывать и получать данные посредством JavaScript, как при использовании стандартного сокета TCP.

URL WebSocket начинается с ws:// или wss:// (по SSL).

Временная диаграмма на рисунке иллюстрирует связь с использованием WebSockets. На сервер посылается HTTP-рукопожатие с определенными заголовками. Затем создается нечто вроде сокета на JavaScript на стороне сервера или клиента. Этот сокет можно использовать для асинхронного приема данных через обработчик событий.

**Рисунок 1. Reverse Ajax с WebSockets**  


**Changelog**

* 010.02.16 - Added basic demo and chatbot
* 010.02.16 - Added users list to keep track of handshakes
* 010.02.16 - Organized everything in a reusable websocket class
* 010.02.16 - Minor cosmetic changes

**Client side**

var host = "ws://localhost:12345/websocket/server.php";  
try{  
  socket = new WebSocket(host);  
  log('WebSocket - status '+socket.readyState);  
  socket.onopen    = function(msg){ log("Welcome - status "+this.readyState); };  
  socket.onmessage = function(msg){ log("Received: "+msg.data); };  
  socket.onclose   = function(msg){ log("Disconnected - status "+this.readyState); };  
}  
catch(ex){ log(ex); }

[View source code for the client](http://code.google.com/p/phpwebsocket/source/browse/trunk/%20phpwebsocket/client.html)

**Server side**

log("Handshaking...");  
list($resource,$host,$origin) = getheaders($buffer);  
$upgrade = "HTTP/1.1 101 Web Socket Protocol Handshake\r\n" .  
           "Upgrade: WebSocket\r\n" .  
           "Connection: Upgrade\r\n" .  
           "WebSocket-Origin: " . $origin . "\r\n" .  
           "WebSocket-Location: ws://" . $host . $resource . "\r\n" .  
           "\r\n";  
$handshake = true;  
socket\_write($socket,$upgrade.chr(0),strlen($upgrade.chr(0)));

[View source code for the server](http://code.google.com/p/phpwebsocket/source/browse/trunk/%20phpwebsocket/server.php)

**Steps to run the test:**

1. Save both files, client.php and server.php, in a folder in your local server running Apache and PHP.
2. From the command line, run the server.php program to listen for socket connections.
3. Open Google Chrome (dev build) and point to the client.php page
4. Done, your browser now has a full-duplex channel with the server.
5. Start sending commands to the server to get some responses.

Source Links

* <https://code.google.com/p/phpwebsocket/>

C# Web Sockets

* <http://www.codeproject.com/Articles/57060/Web-Socket-Server>

Java Web Sockets

* <http://habrahabr.ru/post/128380/>
* <http://docs.oracle.com/javaee/7/tutorial/doc/websocket.htm>
* <http://java.dzone.com/articles/creating-websocket-chat> -- Доступны примеры
* <http://jaxenter.com/tutorial-pushing-browser-updates-using-websockets-in-glassfish-46817.html>
* <http://playjava.wordpress.com/2013/07/11/websocket-in-java-part-1-before-java-ee7/>
* <https://netbeans.org/kb/docs/javaee/maven-websocketapi.html>
* <http://www.byteslounge.com/tutorials/java-ee-html5-websocket-example>
* <http://docs.oracle.com/javaee/7/tutorial/doc/websocket.htm>

**Task**

Apply web sockets to current **blog project** using PHP and one another language.