

Chat - Fundamentals - Live

```
<meta http-equiv="refresh" content="5">
    Read and write to a text file - some server side language

<?
    if (isset($_GET['message'])) {
        file_put_contents("messages.txt", $_GET['message'] . "\n", FILE_APPEND);
    }

    $messages = file_get_contents("messages.txt");
    $messages_array = explode("\n", $messages);
?>
<html>
    <head>
        <meta http-equiv="refresh" content="5">
    </head>
    <body>
        <div>
            <?
                foreach ($messages_array as $message) {
                    echo($message . "<br />");
                }
            ?>
        </div>
        <div>
            <form action="meta_refresh_chat.php" method="get">
                <input type="text" name="message">
                <input type="submit" name="submit" value="submit">
            </form>
        </div>
    </body>
</html>
```

AJAX - Asynchronous JavaScript and XML

XMLHttpRequest - <https://en.wikipedia.org/wiki/XMLHttpRequest>
Security - https://en.wikipedia.org/wiki/Same_origin_policy

Same as above but background/asynchronous HTTP request
Callback

```
<html>
    <head>
        <!-- Load up the AJAX External JavaScript file -->
        <script language="JavaScript" src="ajax.js" />

        <!-- Local Javascript Functions and so on -->
        <script type="text/javascript">

            // A variable to hold the interval id
            var interval = null;

            // A function to call our AJAX PHP script
            function call_ajax()
            {
                makeHttpRequest('messages.txt', ajax_return);
            }

            // A function that gets called when ajax returns
            function ajax_return(response)
            {
                document.getElementById("messages").innerHTML = response;
                // Repeat in 5 seconds
                interval = setTimeout("call_ajax()", 5000);
            }

            // Setup AJAX function, creates a timeout so that we run something periodically
            function setup_ajax()
            {
                // Only happens once..
                interval = setTimeout("call_ajax()", 5000);
            }

            // Register setup_ajax with the onload event of the window (when it is done loading)..
            window.onload = setup_ajax;

        </script>
    </head>
    <body>
        <div id="messages">
            No Messages Yet
        </div>
        <div>
            <form action="save_message.php" method="get">
                <input type="text" name="message">
            </form>
        </div>
    </body>
</html>
```

```

        <input type="submit" name="submit" value="submit">
    </form>
</div>
</body>
</html>

```

save_message.php

```

<?
    if (isset($_GET['message'])) {
        file_put_contents("messages.txt", $_GET['message'] . "\n", FILE_APPEND);
    }

    $messages = file_get_contents("messages.txt");
    $messages_array = explode("\n", $messages);
?>

```

Google, Gmail, Maps, etc...

Drawbacks - Search Engines, Back Button, Lots of requests

jQuery AJAX - Simplified

Commet

[https://en.wikipedia.org/wiki/Comet_\(programming\)](https://en.wikipedia.org/wiki/Comet_(programming))

Aaron Swartz

A Brief History of Ajax

<http://www.aaronsw.com/weblog/ajaxhistory>

2:00 - WebSocket

No real relationship to HTTP other than the setup

https://en.wikipedia.org/wiki/WebSocket#WebSocket_protocol_handshake

TCP Socket - Port 80 (other relationship)

Demos

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

```

<html>
<head>
<script type="text/javascript">
    var websocket;

    var init = function() {
        websocket = new WebSocket("ws://echo.websocket.org");

        // Callbacks
        websocket.onopen = function(evt) { alert("onopen " + evt) };
        websocket.onclose = function(evt) { alert("onclose " + evt) };
        websocket.onmessage = function(evt) { alert("onmessage " + evt.data) };
        websocket.onerror = function(evt) { alert("onerror " + evt) };
    };

    var send = function(message) {
        websocket.send(message);
    };

    window.addEventListener("load", init, false);
</script>
</head>
<body>
<input type="button" value="send" onclick="send();" />
</body>
</html>

```

Need a server that supports it..

Node.js

JavaScript engine for building server side apps - particularly HTTP

<http://nodejs.org/>

AWS - EC2

<http://aws.amazon.com/>

Sign up

Need CC and Phone

Micro Instance/Free Tier - 750 hours

<http://aws.amazon.com/free/>

Console

<https://console.aws.amazon.com/console/home?#>

Launch instance - new key pair - Download

Follow instructions for connect - move key pair

`mv liveweb-keypair.pem ~/.ssh/`

```
chmod 400 ~/.ssh/liveweb-keypair.pem
ssh -i ~/.ssh/liveweb-keypair.pem ec2-user@ec2-54-242-249-213.compute-1.amazonaws.com
whoami
```

Get Node Installed

<http://iconof.com/blog/how-to-install-setup-node-js-on-amazon-aws-ec2-complete-guide/>

We'll be using yum, the Yellowdog Updater, Modified which helps us immensely with installing and keeping software up to date on Linux.

```
sudo yum update
sudo yum install gcc-c++ make
sudo yum install openssl-devel
sudo yum install git
git clone git://github.com/joyent/node.git
cd node

git checkout v0.10.18-release (This may be incorrect)
./configure
make
sudo make install
```

Now to build a Web Socket Demo with socket.io

<http://socket.io/#how-to-use>

```
Let's create a directory to house our demo server
mkdir ~/socketserver_test
cd ~/socketserver_test
```

Install Node module

```
sudo /usr/local/bin/npm install socket.io -g
```

Give it a test

Create server.js on your local machine
Create index.html ...

Command Line to upload

```
sftp -i ~/.ssh/liveweb-keypair.pem ec2-user@ec2-54-242-249-213.compute-1.amazonaws.com
```

```
put server.js
put index.html
(remember to change localhost to your amazon hostname.)
```

Install the socket.io module

```
/usr/local/bin/npm install socket.io
```

Run it

```
sudo /usr/local/bin/node server.js
```

A Chat Example

```
server.js
```

```
var app = require('http').createServer(handler)
    , io = require('socket.io').listen(app)
    , fs = require('fs')
```

```
app.listen(80);
```

```
function handler (req, res) {
  fs.readFile(__dirname + '/index.html',
    function (err, data) {
      if (err) {
        res.writeHead(500);
        return res.end("Error loading index.html");
      }

```

```
      res.writeHead(200);
      res.end(data);
    });
}
```

```
io.sockets.on('connection', function (socket) {
  socket.on('message', function (data) {
    console.log("message: " + data);
    socket.broadcast.emit('message', data);
  });
});
```

```
index.html
```

```
<html>
  <head>
    <script src="/socket.io/socket.io.js"></script>
    <script>
      var socket = io.connect('http://ec2-54-242-249-213.compute-1.amazonaws.com');
      socket.on('news', function (data) {
        console.log(data);

```

```
    });
    socket.on('message', function(data) {
      console.log("Got: " + data);
      document.getElementById('messages').innerHTML += data;
    });

    var sendmessage = function() {
      var message = document.getElementById('message').value;
      console.log("Sending: " + message);
      socket.send(message);
    };

    </script>
</head>
<body>
  <div id="messages">
    No Messages Yet
  </div>
  <input type="text" id="message" name="message">
  <input type="submit" value="submit" onclick="sendmessage();">
</body>
</html>
```