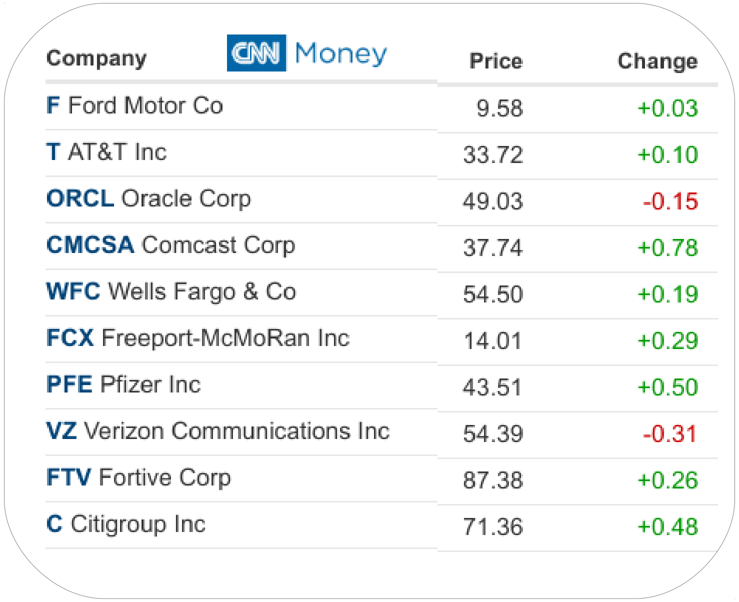
WebSockets vs AJAX

Polling vs Event Driven

- AJAX (REST model) must poll the server
    - Can miss events between polling requests
    - Can be unnecessary if event hasn't happened
- WebSocket is event driven from client and server
    - Either side can initiate xfer to the other independently

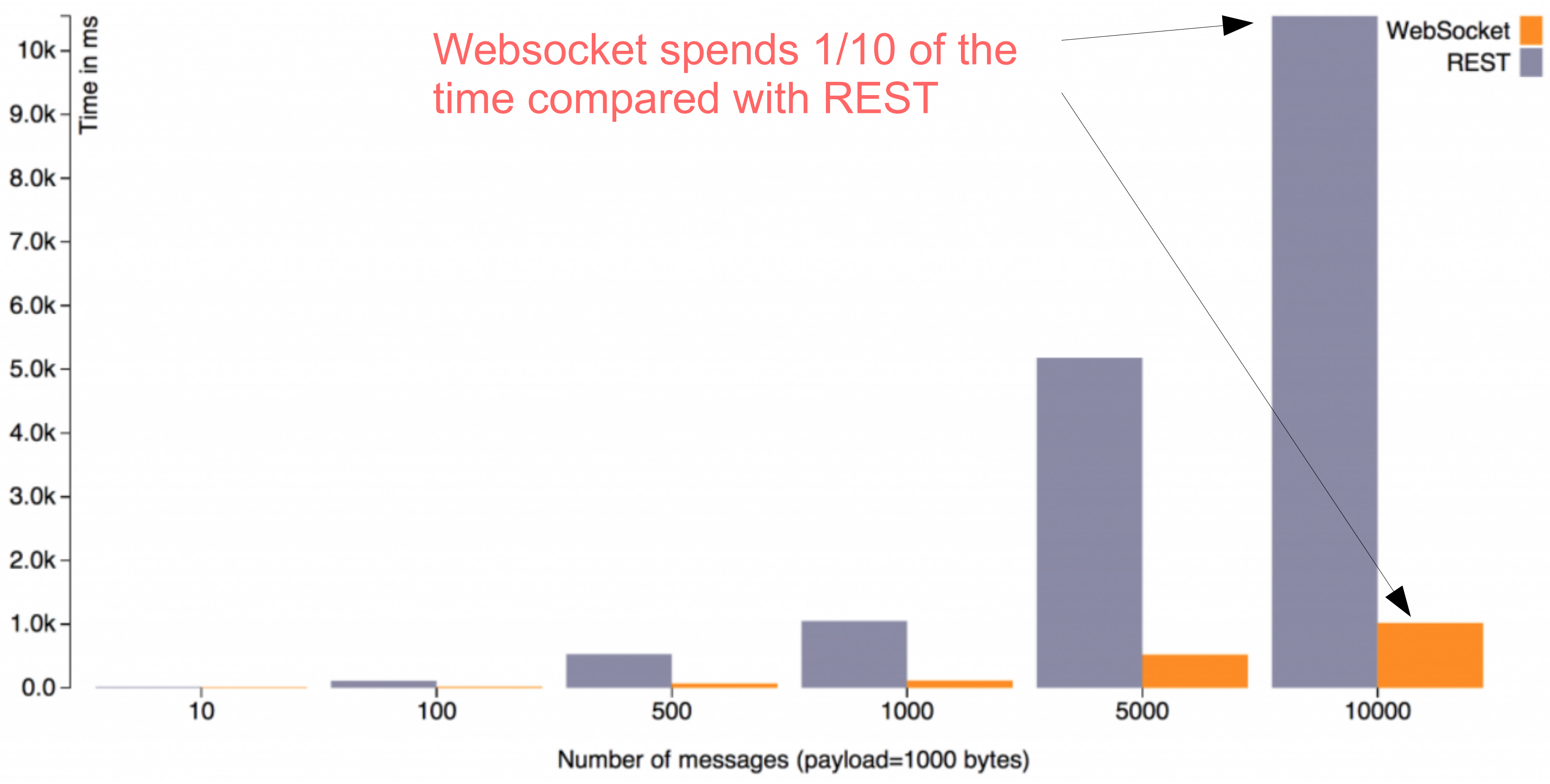
Overhead

- HTTP: New TCP connection for each request
- WebSocket: Single TCP connection, then reused



Company	Price	Change
F Ford Motor Co	9.58	+0.03
T AT&T Inc	33.72	+0.10
ORCL Oracle Corp	49.03	-0.15
CMCSA Comcast Corp	37.74	+0.78
WFC Wells Fargo & Co	54.50	+0.19
FCX Freeport-McMoRan Inc	14.01	+0.29
PFE Pfizer Inc	43.51	+0.50
VZ Verizon Communications Inc	54.39	-0.31
FTV Fortive Corp	87.38	+0.26
C Citigroup Inc	71.36	+0.48

Performance Comparison



Websocket spends 1/10 of the time compared with REST

Source: http://blog.arungupta.me/rest-vs-websocket-comparison-benchmarks/

Super Simple WebSocket Example

github.com/tgregoneil/SuperSimpleWS

Server

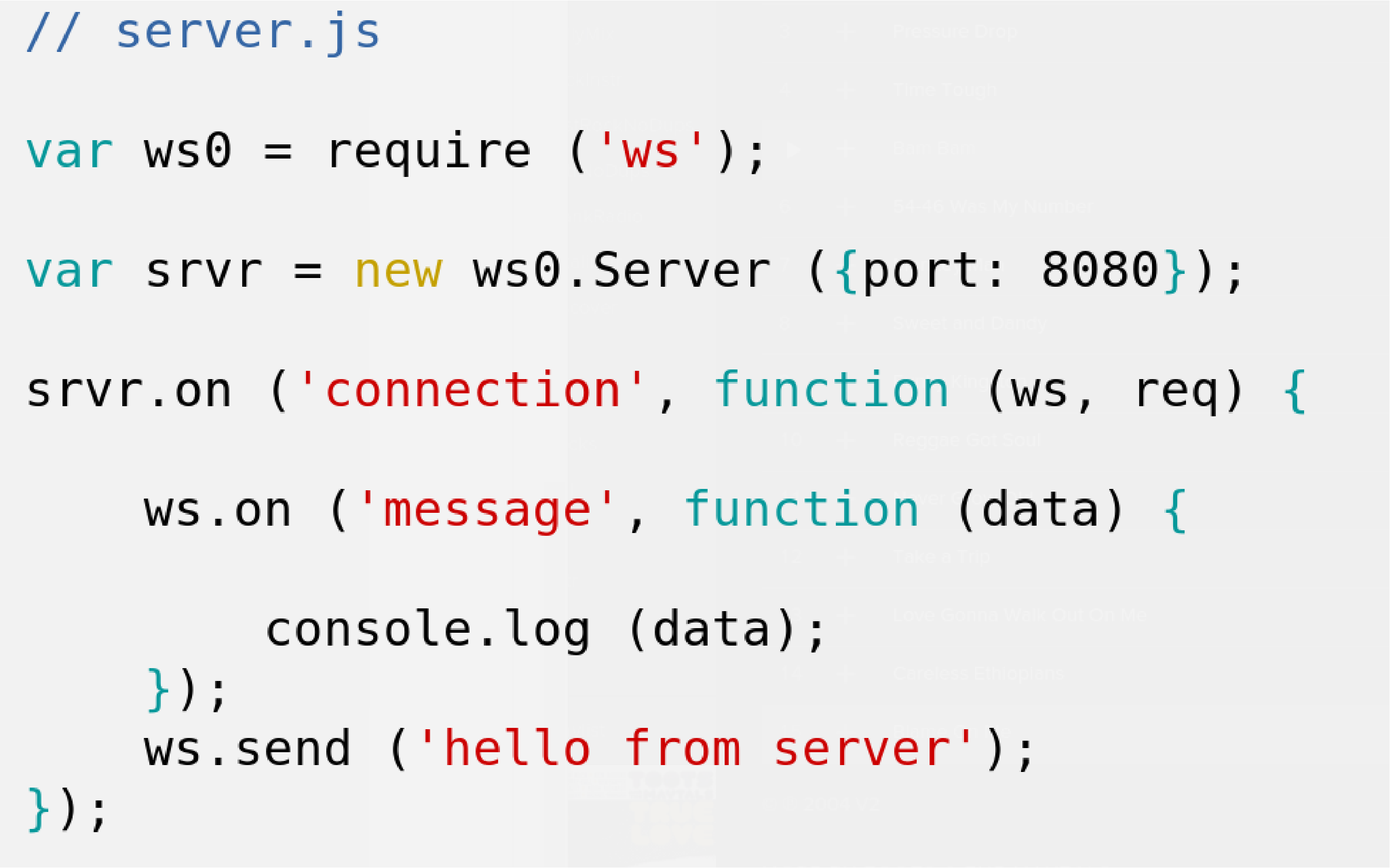
$ npm install ws

$ node server.js

Client

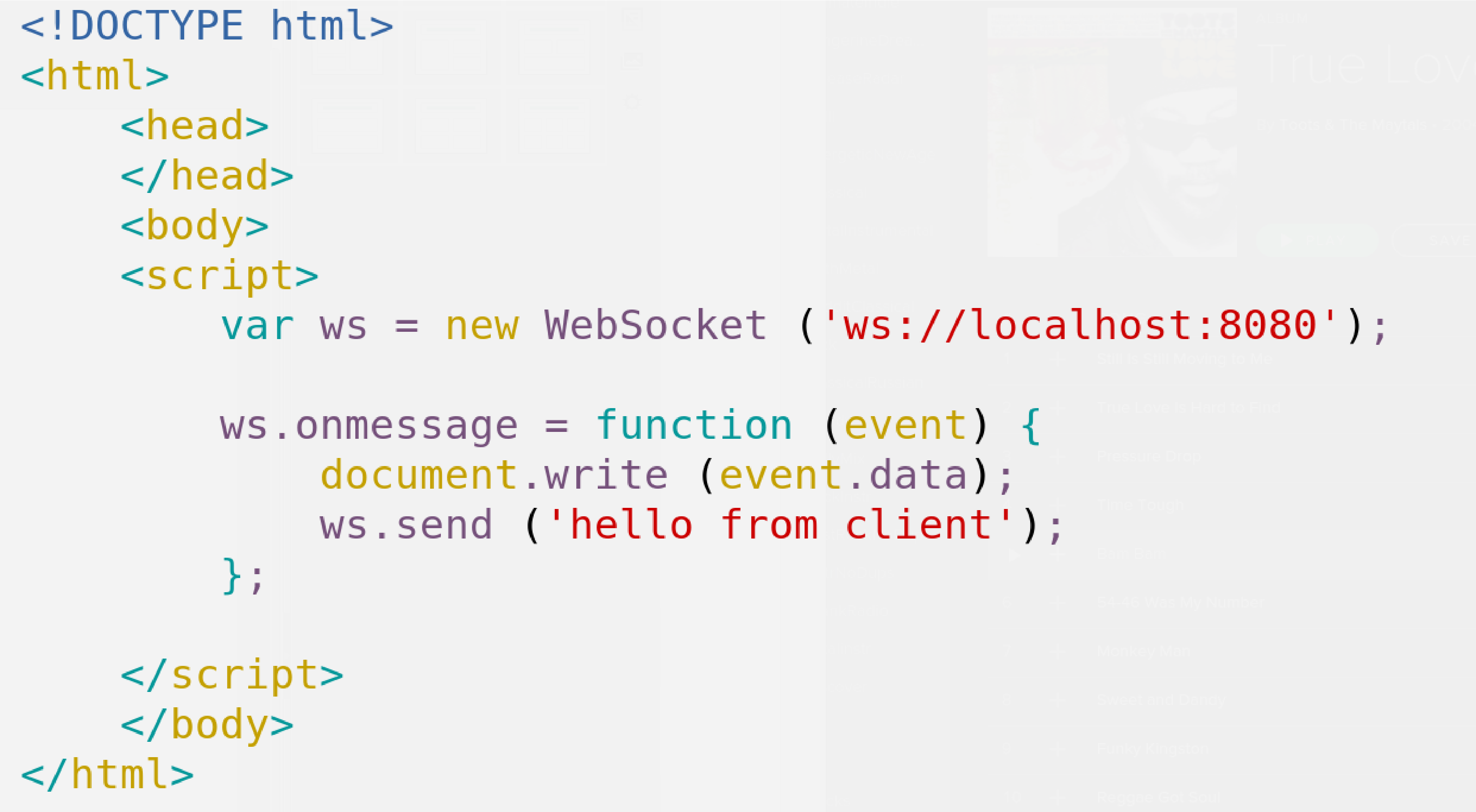
Browse to SuperSimpleWS/index.html

Server



```
// server.js

var ws0 = require ('ws');

var srvr = new ws0.Server ({port: 8080});

srvr.on ('connection', function (ws, req) {

    ws.on ('message', function (data) {

        console.log (data);
    });
    ws.send ('hello from server');
});
```

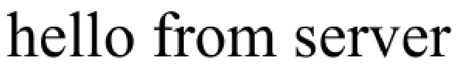
Client



```
<!DOCTYPE html>
<html>
    <head>
    </head>
    <body>
    <script>
        var ws = new WebSocket ('ws://localhost:8080');

        ws.onmessage = function (event) {
            document.write (event.data);
            ws.send ('hello from client');
        };

    </script>
    </body>
</html>
```

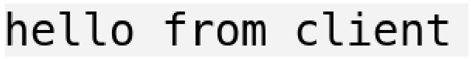
Outputs via Screenshots

Client Browser Window



hello from server

Server Terminal



```
hello from client
```

Chat Server Example

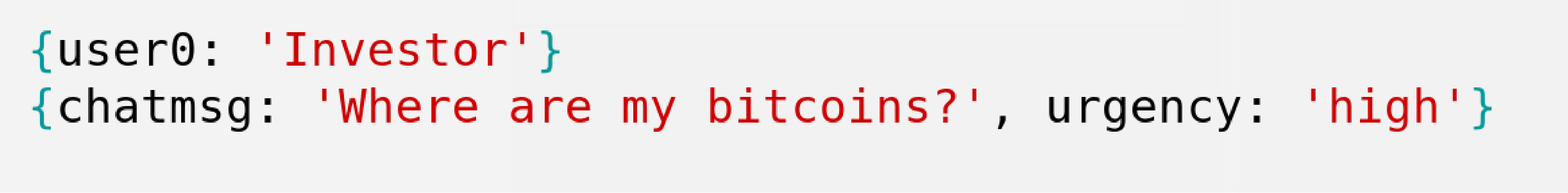
github.com/tgregoneil/go-chat

```
$ npm install go-chat
$ cd Server
$ node chatServer.js
```

Browse to Client/index.html

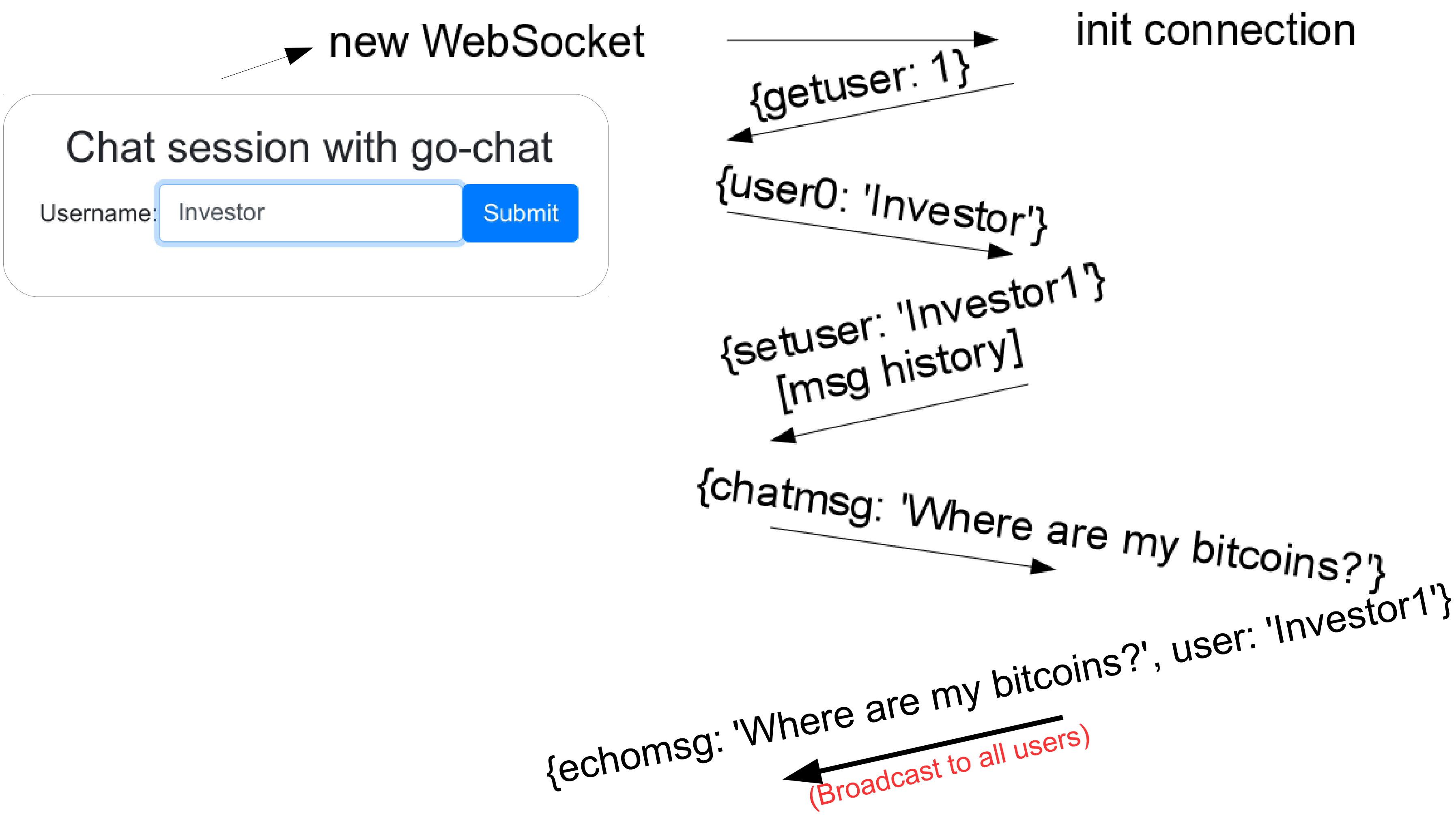
Message Structure

- Object based
  - Each object contains one primary cmd key and optional secondary attributes
  - Stringify object before sending out
  - Parse object on receipt
  - Handle each command
- Example



```
{user0: 'Investor'}
{chatmsg: 'Where are my bitcoins?', urgency: 'high'}
```

Chat Server Design



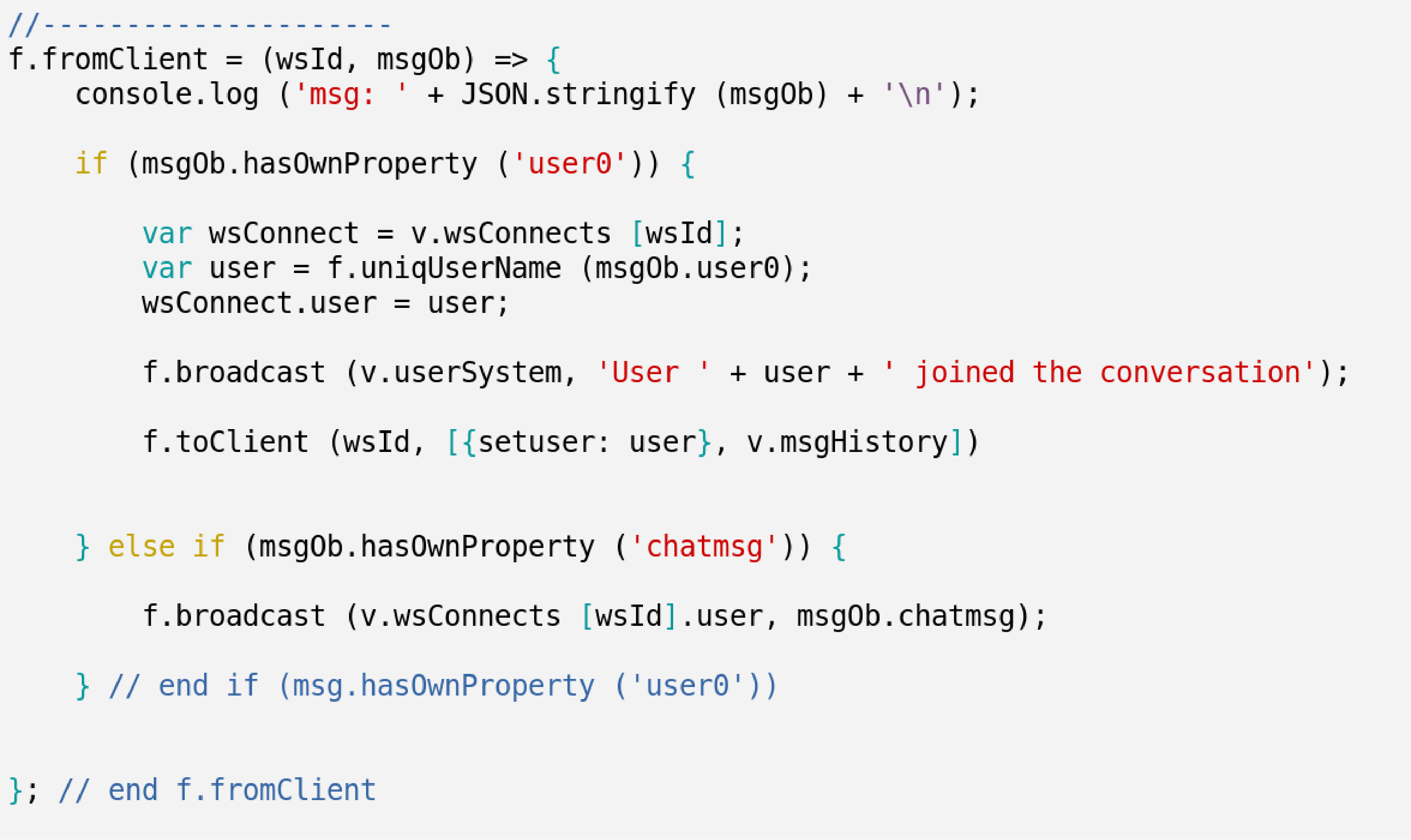
Browser

Server

Chat session with go-chat

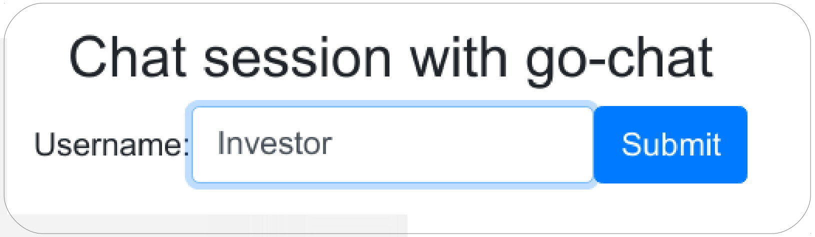
Username: Investor

Handle messages: Server



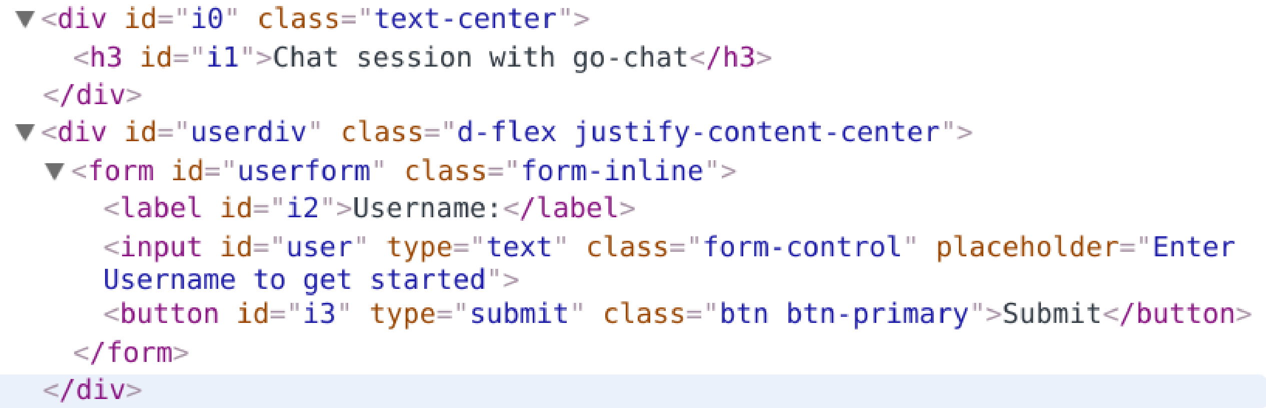
```
//---------------------
f.fromClient = (wsId, msgOb) => {
    console.log ('msg: ' + JSON.stringify (msgOb) + '\n');

    if (msgOb.hasOwnProperty ('user0')) {

        var wsConnect = v.wsConnects [wsId];
        var user = f.uniqUserName (msgOb.user0);
        wsConnect.user = user;

        f.broadcast (v.userSystem, 'User ' + user + ' joined the conversation');

        f.toClient (wsId, [{setuser: user}, v.msgHistory])


    } else if (msgOb.hasOwnProperty ('chatmsg')) {

        f.broadcast (v.wsConnects [wsId].user, msgOb.chatmsg);

    } // end if (msg.hasOwnProperty ('user0'))


}; // end f.fromClient
```

Browser Rendering with go-j2h



Chat session with go-chat

Username: Investor



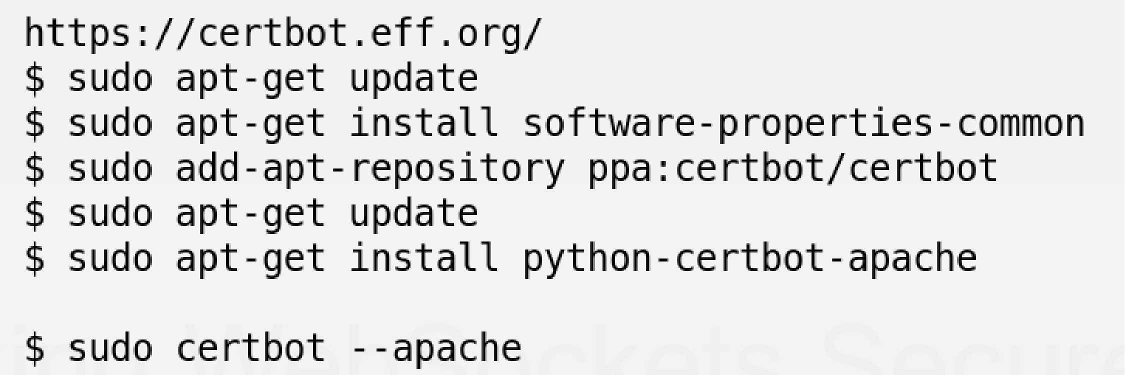
```
▼<div id="i0" class="text-center">
   <h3 id="i1">Chat session with go-chat</h3>
 </div>
▼<div id="userdiv" class="d-flex justify-content-center">
  ▼<form id="userform" class="form-inline">
     <label id="i2">Username:</label>
     <input id="user" type="text" class="form-control" placeholder="Enter
     Username to get started">
     <button id="i3" type="submit" class="btn btn-primary">Submit</button>
   </form>
 </div>
```

```
var header = {
    div: {h3: 'Chat session with go-chat'},
    class: 'text-center',
};

var user = {
    div: {
        form: [
            {label: 'Username:'},
            {
                input: 0,
                id: v.idUserName,
                type: 'text',
                class: 'form-control',
                placeholder: 'Enter Username to get started',
            },
            {button: 'Submit', type: 'submit', class: 'btn btn-primary'},
        ],
        id: v.idUserForm,
        class: 'form-inline',
    },
    id: v.idUserDiv,
    class: 'd-flex justify-content-center',
};

v.j2h ([header, user]);
```

```
$(v.IdUserForm)
.submit (function (event) {
    event.preventDefault();
    v.user0 = $(v.IdUserName).val();
         ...
    v.wsServer = new WebSocket (urlWs);
```

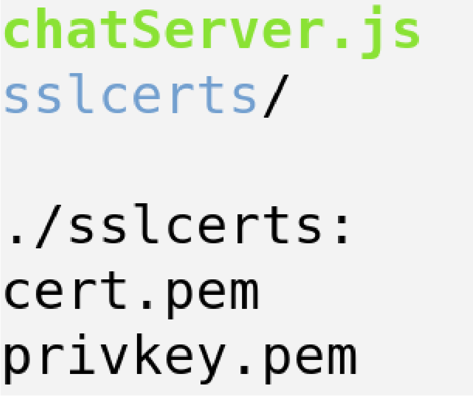
Making WebSockets Secure using https certs

=> LetsEncrypt is free if site does not yet support https



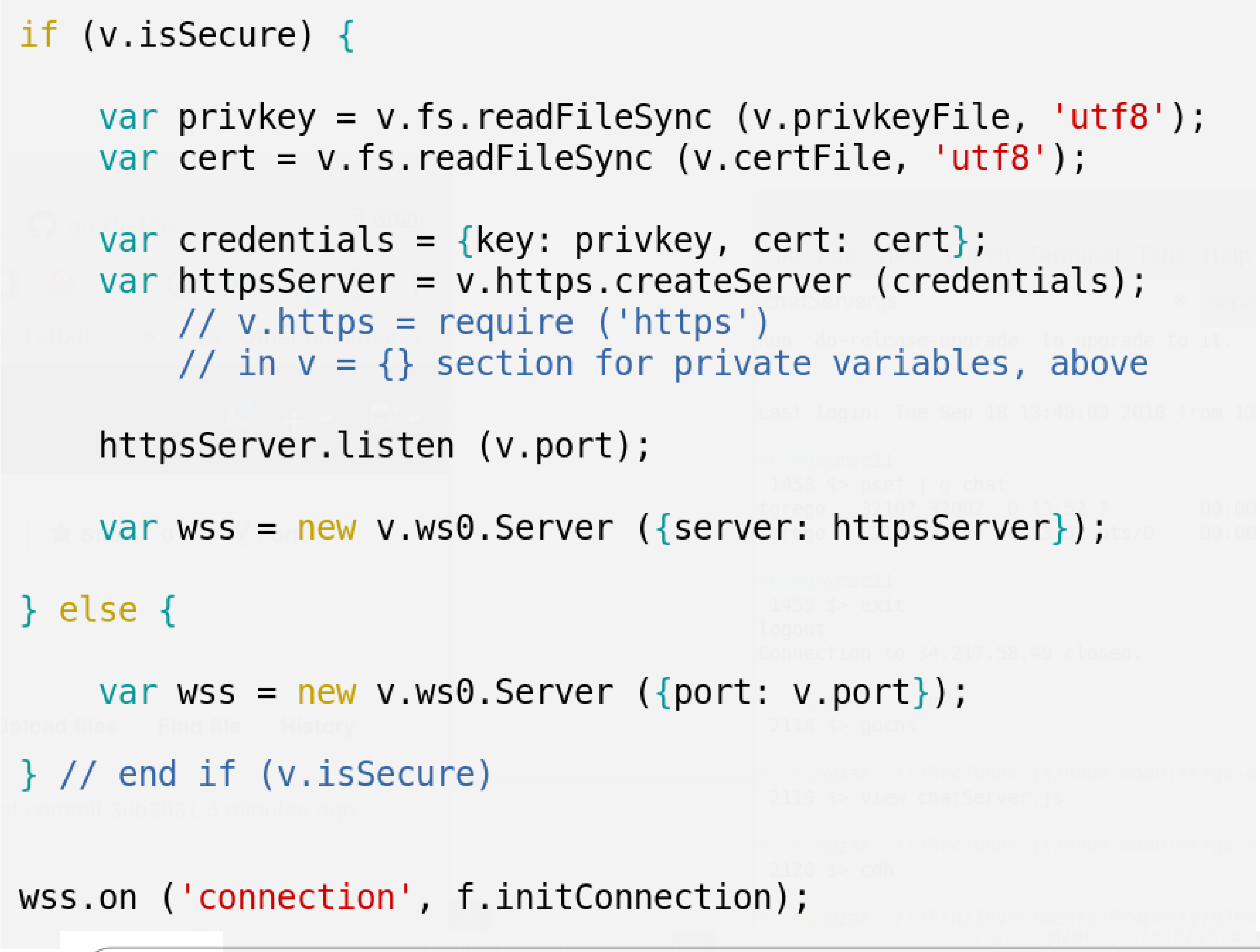
```
https://certbot.eff.org/
$ sudo apt-get update
$ sudo apt-get install software-properties-common
$ sudo add-apt-repository ppa:certbot/certbot
$ sudo apt-get update
$ sudo apt-get install python-certbot-apache

$ sudo certbot --apache
```

Copy or link certs to Server directory



```
chatServer.js
sslcerts/

./sslcerts:
cert.pem
privkey.pem
```

Secure WebSocket Server Code



```
if (v.isSecure) {

    var privkey = v.fs.readFileSync (v.privkeyFile, 'utf8');
    var cert = v.fs.readFileSync (v.certFile, 'utf8');

    var credentials = {key: privkey, cert: cert};
    var httpsServer = v.https.createServer (credentials);
        // v.https = require ('https')
        // in v = {} section for private variables, above

    httpsServer.listen (v.port);

    var wss = new v.ws0.Server ({server: httpsServer});

} else {

    var wss = new v.ws0.Server ({port: v.port});

} // end if (v.isSecure)


wss.on ('connection', f.initConnection);
```

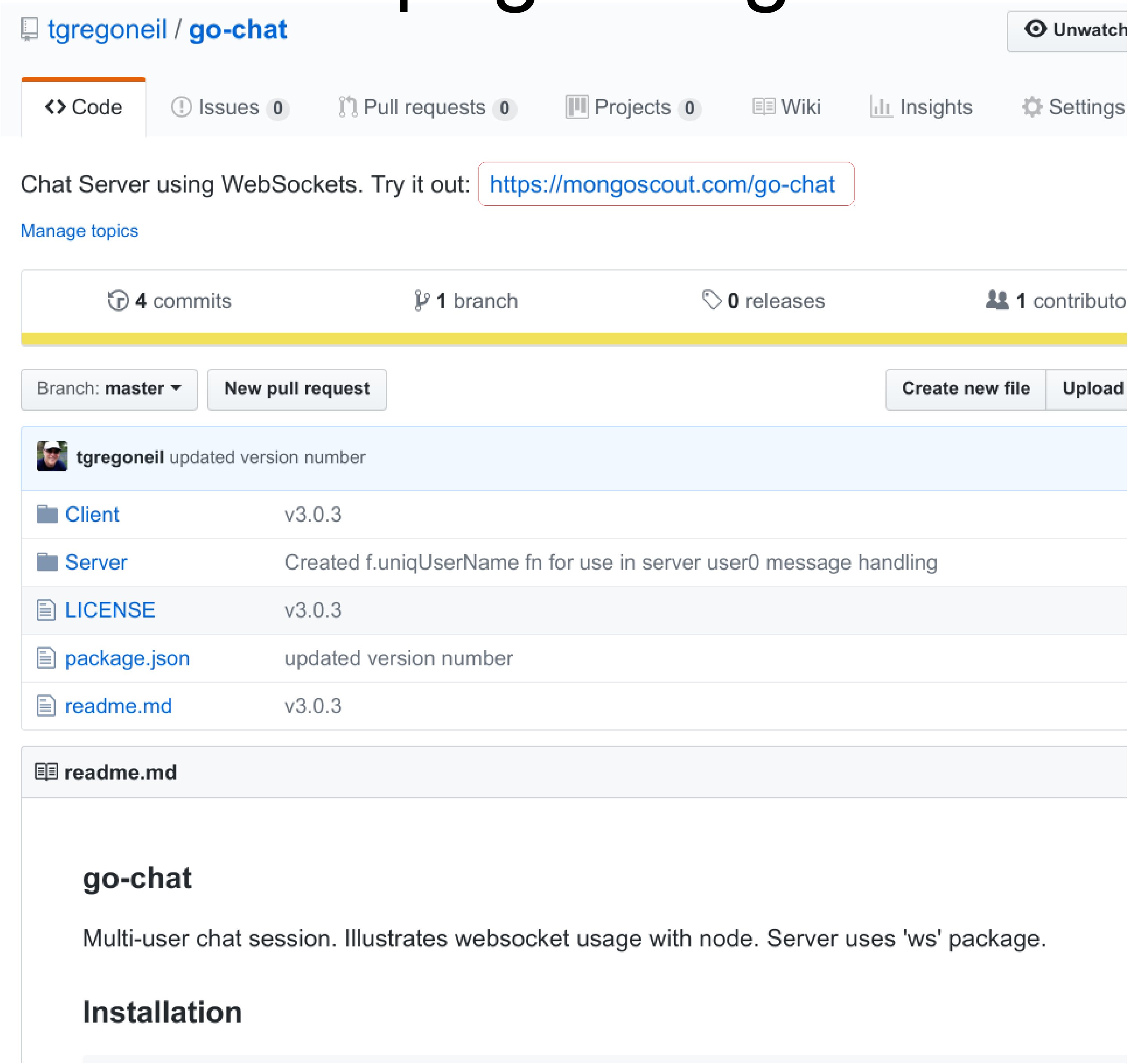
From Server directory, invoke chat server with '--secure' option:

```
$ node chatServer.js --secure
```

Secure WebSocket Browser Code

```
var url = document.URL;

var isHttps = false;
var domain = 'localhost';

var matched = url.match (/(https?):..([^\/]+)/);
    // e.g. https://mongoscout.com/go-chat

if (matched) {

    isHttps = matched [1] === 'https';
    domain = matched [2];

} // end if (matched)

var prefix = isHttps ? 'wss' : 'ws';

var urlWs = prefix + '://' + domain + ':' + v.port;
v.wsServer = new WebSocket (urlWs);
```

Github page for go-chat



tgregoneil / go-chat

Chat Server using WebSockets. Try it out: https://mongoscout.com/go-chat

Manage topics

4 commits 1 branch 0 releases 1 contributo

tgregoneil updated version number

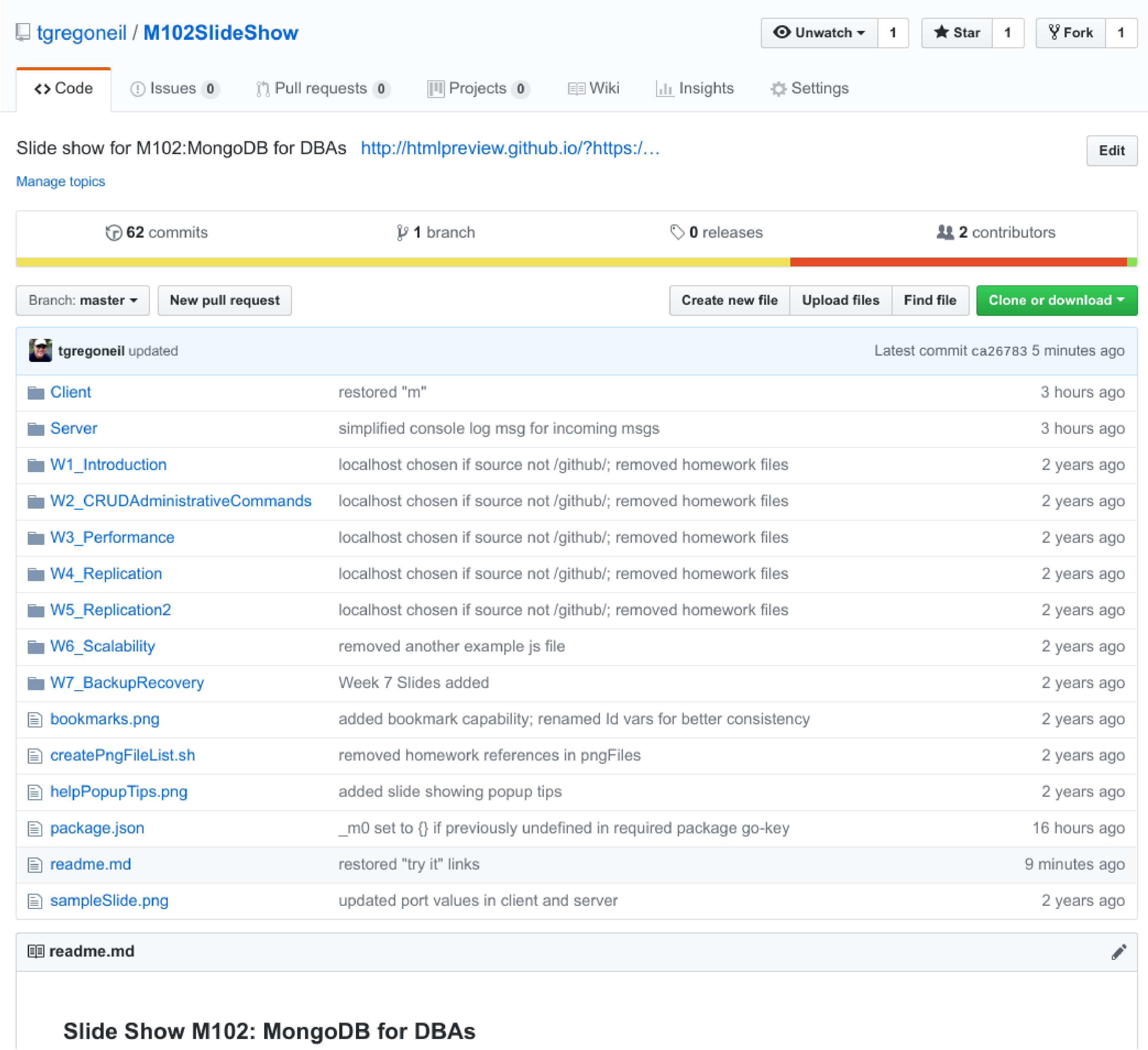
Client	v3.0.3
Server	Created f.uniqUserName fn for use in server user0 message handling
LICENSE	v3.0.3
package.json	updated version number
readme.md	v3.0.3

readme.md

go-chat

Multi-user chat session. Illustrates websocket usage with node. Server uses 'ws' package.

Installation



tgregoneil / **M102SlideShow**

Slide show for M102:MongoDB for DBAs http://htmlpreview.github.io/?https:/…

Manage topics

62 commits | **1** branch | **0** releases | **2** contributors

tgregoneil updated — Latest commit ca26783 5 minutes ago

Client	restored "m"	3 hours ago
Server	simplified console log msg for incoming msgs	3 hours ago
W1_Introduction	localhost chosen if source not /github/; removed homework files	2 years ago
W2_CRUDAdministrativeCommands	localhost chosen if source not /github/; removed homework files	2 years ago
W3_Performance	localhost chosen if source not /github/; removed homework files	2 years ago
W4_Replication	localhost chosen if source not /github/; removed homework files	2 years ago
W5_Replication2	localhost chosen if source not /github/; removed homework files	2 years ago
W6_Scalability	removed another example js file	2 years ago
W7_BackupRecovery	Week 7 Slides added	2 years ago
bookmarks.png	added bookmark capability; renamed Id vars for better consistency	2 years ago
createPngFileList.sh	removed homework references in pngFiles	2 years ago
helpPopupTips.png	added slide showing popup tips	2 years ago
package.json	_m0 set to {} if previously undefined in required package go-key	16 hours ago
readme.md	restored "try it" links	9 minutes ago
sampleSlide.png	updated port values in client and server	2 years ago

readme.md

Slide Show M102: MongoDB for DBAs

Parting Thoughts

- Drink Blended Organic Green Smoothies every day
- Index-based mutual funds and ETFs outperform 75-80% of actively managed counterparts
- Long live Bootstrap and especially jQuery!