

Creating a Socket.IO chat application

Estimated time for completion: 30 minutes

Overview:

In this lab you will use Node.js and the Socket.IO framework to create a simple multi user chat application. The chat application will use WebSockets as the communications protocol if possible add fall back on another protocol if needed.

Goals:

* Use Socket.IO to create a chat server
* Use Socket.IO and jQuery to create a chat client

Lab Notes:

The lab uses Node.js.

* You can install Node.js from [http://nodejs.org](http://nodejs.org/)

Creating the Chat Server

In this part of the lab, you will implement the chat server using Socket.IO.

Criteria:

* Modify the Server.js to allow users to chat using a chat server implemented using Socket.IO.

Steps:

1. This lab is meant to be a very basic introduction to using the Socket.IO framework for direct socket style communications over the internet. You will be using the popular Node.js Express package to serve the static files the client application needs.
2. Open the Server.js from ~/SocketIO/before. In this file you will add the basic chat server.
3. Add Socket.IO and Express to the project using the Node Packet Manager, NPM, and require them in Server.js.
4. Use Express to server all static files located in the Public folder. Make sure to return the Default.html when the root of the site is requested.
5. Using Socket.IO listen for users connecting and support the following API:
   1. The user connecting  
      Broadcast the user connecting to all other users.
   2. User setting their user name.  
      Save the user name with the connection for broadcasting updates.
   3. Users sending an chat update  
      Broadcast the update to all other users with the originating users name.
   4. The user disconnecting  
      Notify all other users of the disconnecting user.
6. Make sure the service starts without problems. In the next section you will implement the client to test the actual functionality.

**Helpful links:**

* + [Getting started with Socket.IO](http://socket.io/#how-to-use)
  + Also contains a section on how to use the Express 3 web framework
  + Use express.static() to host static files in [Express.js](http://expressjs.com/api.html)

Creating an HTML/JavaScript/Node.IO chat client

In this part of the lab you will be updating the client by adding the required client side Node.IO code to the web page.

Criteria:

* Be able to send updates to all other connected clients.
* Receive chat updates from all other connected clients.

Steps:

* Open the file Default.html and inspect the general layout of the client application.
* Open the App.js file. This file is currently just a boilerplate with ToDo reminders and you will be adding the code to make the chat client functional.

**Helpful links:**

* + [Getting started with Socket.IO](http://socket.io/#how-to-use)
  + [Using HTML5 Web Storage](http://www.w3schools.com/html/html5_webstorage.asp)
  + [Using jQuery](http://jquery.com/)

Solutions:

The final solution for this lab is available in the ~/after directory.