リアルタイム更新チャット

2016年7月29日

倉地　将吾

リアルタイム更新チャットに関することをします。

WebSocketを扱うことができるNode.jsのライブラリ「Socket.IO」を使って、サンプルアプリケーションを構築していきます。

１．Node.jsのインストール

下記のサイトを参考にNode.jsをインストールしてください。

サイト名：Node.js / npmをインストールする（for Windows）

URL ：http://qiita.com/krtbk1d/items/9001ae194571feb63a5e

２．Socket.IOのインストール

下記のサイトを参考にSocket.IOをインストールしてください。

サイト名：第1回　いまさら聞けないWebSocketとSocket.IOの基礎知識＆インストール

URL ：http://www.atmarkit.co.jp/ait/articles/1603/14/news015.html

手順１：デスクトップ上に「websocket-chat」ディレクトリを作成してください。

手順２：コマンドプロンプトで「websocket-chat」ディレクトリへ移動してください。

手順３：コマンドプロンプトで「npm install socket.io」のコマンドを実行してください。

手順４：「websocket-chat」ディレクトリ配下に「node\_modules/socket.io」ディレクトリが配置されていればSocket.IOのインストールは完了です。

３．チャットアプリケーションのひな型作成

手順１：「websocket-chat」ディレクトリ内に「index.html」「app.js」

手順２：index.html：クライアントサイド（ユーザーが使用するチャット画面を実装）

手順３：app.js：サーバサイド（メッセージ送受信等サーバサイド処理を実装）

index.html

<!DOCTYPE html>

<html lang="ja">

<head>

<meta charset="utf-8">

<title>websocket-chat</title>

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>

<script type="text/javascript" src="/socket.io/socket.io.js"></script>

</head>

<body>

<div class="container">

<h1>WebSocket-Chat</h1>

<form class="form-inline">

<div class="form-group">

<label for="msgForm">メッセージ：</label>

<input type="text" class="form-control" id="msgForm">

</div>

<button type="submit" class="btn btn-primary">送信</button>

</form>

<div id="chatLogs"></div>

</div>

<script type="text/javascript">

var socket = io.connect();

socket.on("server\_to\_client", function(data){appendMsg(data.value)});

function appendMsg(text) {

$("#chatLogs").append("<div>" + text + "</div>");

}

$("form").submit(function(e){

var message = $("#msgForm").val();

$("#msgForm").val('');

socket.emit("client\_to\_server", {value : message});

e.preventDefault();

});

</script>

</body>

</html>

app.js

var http = require('http');

var socketio = require('socket.io');

var fs = require('fs');

var server = http.createServer(function(req, res) {

res.writeHead(200, {'Content-Type' : 'text/html'});

res.end(fs.readFileSync(\_\_dirname + '/index.html', 'utf-8'));

}).listen(3000); // ポート競合の場合は値を変更

var io = socketio.listen(server);

io.sockets.on('connection', function(socket) {

socket.on('client\_to\_server', function(data) {

io.sockets.emit('server\_to\_client', {value : data.value});

});

});

４．チャットアプリケーションの実行確認

動作確認を行います。

手順１：コマンドプロンプトで「websocket-chat」ディレクトリに移動してください。

手順２：コマンドプロンプトで「node app.js」を実行します。

手順３：「http://localhost:3000/」に2つのウィンドウから接続してください。

終了方法はコマンドプロンプトでCtrl+Cを押下またはコマンドプロンプトを終了してください

５．別々のPCでチャットを行う方法

手順１:チャットアプリケーションを実行しているPCのコマンドプロンプトで「ipconfig」コマンドを実行し、IPアドレスを確認する。

手順２:別のPCで「http://調べたIPアドレス:3000/」に接続する

６．プロジェクトに入れる方法

手順１：実装したいプロジェクトの直下に「websocket-chat」ディレクトリを貼り付け

参考文献

サイト名　：Node.js / npmをインストールする（for Windows）

URL 　：http://qiita.com/krtbk1d/items/9001ae194571feb63a5e

最終閲覧日：2016年7月29日

サイト名：第1回いまさら聞けないWebSocketとSocket.IOの基礎知識＆インストール

URL ：http://www.atmarkit.co.jp/ait/articles/1603/14/news015.html

最終閲覧日：2016年7月29日

サイト名　：第2回　チャットアプリ開発に見る、Socket.IOの基本ライブラリの使い方

URL 　：http://www.atmarkit.co.jp/ait/articles/1604/27/news026.html

最終閲覧日：2016年7月29日

サイト名　：第3回　Socket.IO開発時に役立つツール4選とroom、namespaceライブラリの使い方

URL 　：http://www.atmarkit.co.jp/ait/articles/1607/01/news027.html

最終閲覧日：2016年7月29日

サイト名　：5.1.1　自分のIPアドレスを調べてみる/コマンドライン

URL 　：<http://www.arearesearch.co.jp/learn/program/01.html>

最終閲覧日：2016年7月29日