

Web Application Development

A problem-based approach
Volume II

Gilles Everling

gilles.everling@education.lu

Latest version available at <https://web.cnfi.lu/WebApplicationDevelopment.pdf>

This version created on September 30, 2016

Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

Contents

Contents	ii
1 Project management	1
2 NeoVim	2
2.1 Installation	2
2.2 File explorer, tabs, opening files	2
3 Node.js	3
3.1 Installation and upgrade	3
3.1.1 Ubuntu	3
3.1.2 Windows	3
3.2 Introduction	3
3.3 API	3
3.3.1 Process	3
3.4 async	4
3.5 Sessions	4
3.6 Authentication	4
3.6.1 JSON Web Tokens	4
3.7 Automatic script restart	4
4 WebSocket	5
4.1 Introduction	5
4.2 Server	5
4.2.1 ws	5
4.2.2 Socket.IO	5
4.3 Server	5
4.4 Client	5
5 WebRTC	6
6 Cross platform app development	7
6.1 Electron	7
6.2 NW.js	7

Chapter 1

Project management

Chapter 2

NeoVim

2.1 Installation

`https://github.com/neovim/neovim/wiki/Installing-Neovim`
`nvim :Tutor`

2.2 File explorer, tabs, opening files

`:help explore :newtab gt :Explore :o|`
`https://askubuntu.com/questions/492063/can-i-open-two-or-more-fi`
`les-in-vim-at-once`
`http://vim.wikia.com/wiki/File_explorer`

Chapter 3

Node.js

3.1 Installation and upgrade

3.1.1 Ubuntu

<https://github.com/nodesource/distributions#installation-instructions>

3.1.2 Windows

<https://nodejs.org> <https://www.npmjs.com>

You need to add an environment variable as described in <http://stackoverflow.com/questions/9587665>.

3.2 Introduction

<https://github.com/maxogden/art-of-node>
<https://github.com/substack/stream-handbook>
<http://requirebin.com>
<http://nodeschool.io>
<https://github.com/workshopper/learnyounode>
<http://chimera.labs.oreilly.com/books/123400000262/index.html>
<http://jsbooks.revolunet.com>
`mkdir myapp cd myapp npm init npm i --save express`

3.3 API

3.3.1 Process

From <https://nodejs.org/dist/latest-v6.x/docs/api/process.html>:

The process object is a global that provides information about, and control over, the current Node.js process. As a global, it is always available to Node.js applications without using `require()`.

3.4 async

3.5 Sessions

<https://stormpath.com/blog/everything-you-ever-wanted-to-know-about-node-dot-js-sessions>

3.6 Authentication

3.6.1 JSON Web Tokens

https://en.wikipedia.org/wiki/JSON_Web_Token

JWT should not be used for sessions:

<http://crypto.net/~joepie91/blog/2016/06/13/stop-using-jwt-for-sessions>

Node JSON Web Token implementation (symmetric and asymmetric): <https://www.npmjs.com/package/jsonwebtoken>

<https://github.com/dwyl/learn-json-web-tokens>

JSON Web Algorithms (JWA) <https://tools.ietf.org/html/rfc7518#page-9>

<https://scotch.io/tutorials/authenticate-a-node-js-api-with-json-web-tokens>

JWT with socket.io: <https://auth0.com/blog/2014/01/15/auth-with-socket-io>

Debugger: <https://jwt.io>

3.7 Automatic script restart

nodemon <https://www.npmjs.com/package/nodemon>

Chapter 4

WebSocket

4.1 Introduction

The WebSocket API can be found at <https://html.spec.whatwg.org/multi-page/comms.html#network>. The WebSocket Protocol can be found at <https://tools.ietf.org/html/rfc6455#page-55>.

The WebSocket protocol name registry is at <http://www.iana.org/assignments/websocket/websocket.xhtml>.

Overview of the main WebSocket packages for Node: <http://stackoverflow.com/questions/16392260/which-websocket-library-to-use-with-node-js>

4.2 Server

4.2.1 ws

We will use ws (<https://www.npmjs.com/package/ws>) as WebSocket implementation in Node.js.

To install the package: `npm install -g ws bufferutil utf-8-validate`

4.2.2 Socket.IO

In your app directory `npm i --save socket.io`

<http://socket.io/get-started>

http://www.tutorialspoint.com/socket.io/socket.io_overview.htm

File upload: <https://github.com/vote539/socketio-file-upload>

4.3 Server

4.4 Client

Chapter 5

WebRTC

<https://codelabs.developers.google.com/codelabs/webrtc-web>

Chapter 6

Cross platform app development

6.1 Electron

<http://electron.atom.io>

6.2 NW.js

<http://nwjs.io>