Connected Dementia Research: Technology Overview

Introduction

This document will provide developers and maintainers with an overview of the technologies used in the project, and descriptions of the project's file structure and its build process.

Technologies used in the project

- **Github**: A service providing remote hosting and management of **git** repositories.
- **NodeJS**: A Javascript command line environment for developing applications that can run outside of a browser environment. The Node Package Managemer (*npm*) repository is rich source of prebuilt packages containing modules that can be run in the **NodeJS** environment.
- **Yarn**: A package manager for **NodeJS** packages with secure and reliable dependency management that improves on the features provided by the **npm** application. The packages that are required for a project are defined in a file called *package.json* and stored in the *node_modules* directory
- **Gulp**: A task runner using **NodeJS** modules to accomplish a given task. The tasks are defined in a configuration file called *Gulpfile.js* or *gulpfile.js* and run from the command line using the `gulp` command with the name of the task as a parameter (for instance `gulp build`). Running the command `gulp` with no parameters will run the default task if one has been defined.

  At its most basic, each task is a series of Gulp API calls or calls to any **NodeJS** module declared with a `require()` statement, where the output of one task is piped as the input to the next. Each task usually begins with a call to `src()` defining which input files to use, and ending with a call to `dest()` defining where the output files are generated.
- **Webpack**: A static module bundler for JavaScript applications. It will take one or more Javascript modules and bundle them into a one or more runtime files. When **Webpack** processes the application it will build a dependency graph to ensure it includes each dependent module once and once only, so - for instance - if more than one module requires **jQuery** then it will only be included in the bundle once.

  *[NOTE: While **Gulp** and **Webpack** work in fundamentally different ways, each can accomplish many of the same tasks, but the bundling of Javascript files is more robust with **Webpack**. However*

*configuration beyond bundling Javascript is less intuitive/more complex so I am currently running a simple instance of **Webpack** as part of the `scripts` task within **Gulp**.)]*

- **Nunjucks**: A templating language using Handlebars-style syntax. It is a NodeJS port of the Python-based **Jinja2** templating language. It offers a fairly rich set of templating tags with block inheritance, autoescaping, macros, filtering, etc.

Project structure

Git repository

The git repository contains the following branches:

- **master**: The stable branch containing the current build-ready and deployable code.
- **develop**: The branch containing the code integrating features which have completed development but are not yet fully tested.
- **feature/FEATURE_NAME>**: A feature branch containg code for a feature currently under development, but not yet integrated.
- **dist**: A subtree branch containing only the currently deployable `dist/` folder.

  (NOTE: Once I have fully tested the `git subtree` process, this branch will become the release branch for the built application, and the `dist/` folder will be removed from the master branch.)

Features branches are managed by the **git-flow** workflow.

Top-level file structure

```
app/                    # Source files
dist/                   # Generated web pages
node_modules/           # NodeJS packages
Gulpfile.js             # Task runner configuration
package.json            # NodeJS package configuration
DOCUMENTATION.md        # This documentation file
README.md               # Project README file
server.js               # ????
webpack.config.js       # Javascript bundler configuration
yarn-error.log          # Package manager error log
yarn.lock               # Package manager lockfile
```

app/

This folder is under version control and contains all of the source files required to build the production web pages, except for the `NodeJS` packages managed by `yarn`.

dist/

This folder contains the built web application (HTML,CSS, Javascript and other assets) ready for deployment to the development or production server.

[Note: Ideally this folder should not be under version control as it is completely replaced on every build, resulting in large commits every time, but until I have fully tested the `git subtree` process it will remain under version control in the master and develop branches.]

node_modules/

This folder contains the **NodeJS** packages and dependencies defined in the `package.json` file. Because it completely managed by **Yarn**, and can be reinstalled and updated at any time, it is not under version control.

Gulpfile.js

This file contains the configuration for all of the **Gulp** tasks.

package.json

This file contains the configuration for the **NodeJS** packages that are required by the project, along with metadata about the project. It also defines aliases for some simple command line tasks.

DOCUMENTATION.md and DOCUMENTATION.pdf

The **Markdown** source and the current **PDF** version of this file.