

# How to Setup a Documentation Project

Jason McGinnis

June 22, 2017

## Contents

<b>Create the Documentation Project</b>	<b>3</b>
Install Requirements . . . . .	3
Initialize Project . . . . .	3
Build File Structure . . . . .	5
Header File . . . . .	5
Other Files . . . . .	5

# Technical Documentation

Figure 1: Main logo

# Create the Documentation Project

Summary information

About how markdown is great for documentation

There will be primer sections for each of the tools, like git, pandoc, markdown language and atom

## Install Requirements

The document assumes a specific setup with specific tools. The reader is free to use different tools. However, this approach was designed to be very manageable and efficient.

- Node.js (and NPM)
- (optional) Atom.io
- Atom Beautify (package addon)
- Git Plus (package addon)
- language-markdown (package addon)
- pandoc
- MiKTeX for TeX language and PDF creation
- Sumatra for reading PDF

Note: To install addon packages in Atom.io, go to File>Settings>Packages and search in the filter at the top.

## Initialize Project

The first thing to do is to setup the basic structure. Create a folder and enter that folder. From there, use this command to create a bare git repository structure.

```
git init --bare
```

Now, identify where this repository is going to be located and copy these generated files into the new location.

Create a local documentation folder where you will work on and update this project. From this folder, check out the bare git repository.

Note: When using these utilities, notice that the slashes are all forward facing like Linux or the web. If you are on a Windows machine, be careful not to copy-paste paths without updating the slashes.

Tip: If you need to create a `.gitignore` file in Windows, create a new text file and rename it to `.gitignore`. to bypass the error message.

```
git clone "//server/share/repo"
```

Example

```
c:\> mkdir documentation
c:\> cd documentation
c:\documentation> git clone "//MainServer/Shared/TechDocumentation"
c:\documentation> cd TechDocumentation
c:\documentation\TechDocumentation>
```

The remaining examples will assume that you are located at the root of the cloned directory, unless otherwise stated.

Now, setup the node project and respond to the prompts.

```
npm init
```

Edit the package.json to add a `scripts` section after the `main`. These files do not exist yet, but they serve as an example to how you can build your structure. You can write each section separately and then piece them together by including each one. For example, we can have a header, section1 and section2. The header is in the root folder and the sections are in the sectionName folder.

Most likely there is a test script already in the scripts section of the package file. You can either ignore it or delete it. It is not used in this project, because this is for building documentation.

If you desire, you can create different variations or editions to the documentation piece together different parts for different target audiences. A **build-doc-working** version was added to show this example in building a document for editing or working purposes. In this example the Table of Contents and the header section were both removed.

Also added to the scripts section are the version management scripts. These will allow you to run a command like **npm run ver-major**. All the included options are the major, minor, path and even a version reset.

```
"main": "index.js",
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1",
  "build-doc": "pandoc -o tech-doc-manual.pdf -S -s --toc ./main/header.md ./main/requirements.md ./main/ini",
  "build-other-doc": "pandoc -o tech-doc-manual-editing.pdf -S -s ./sectionName/section1.md ./sectionName/se",
  "ver-major": "npm version major",
  "ver-minor": "npm version minor",
  "ver-patch": "npm version patch",
  "ver-reset": "npm version \"0.0.1\" --force"
},
```

Create a batch file to run the script and build the document. To update the files included in the document, you will need to update the **build-doc** script in the package file. The batch file just makes it easy to have an icon to click to run it. Paste the following command into the new batch file. You can also type the following to run the script.

```
npm run build-doc
```

If you run the batch file and you see an **npm-debug.log** file instead of your document file, most likely one of the files could not be found. Check your filenames and paths.

Add a **.gitignore** to the root structure and put in **node\_modules/\*\*** to ignore installed libraries. These are not needed because the required libraries are specified in your package file. Be sure to install any libraries using the save switch. **npm install --save library-name**

## Build File Structure

Begin buiding the files and folders needed to keep your documentation organized.

### Header File

The first file to create defines the top section of the document.

header.md

```
---
title: Title of Your Documentation
author: Your Name
date: \today
geometry: "top=1in, bottom=.5in, left=.5in, right=.5in"
---
```

\pagebreak

![Main logo](insertLogoHere.jpg){ width=80% }

\pagebreak

# Main Concept

[Link to product](http://link-to-product)

Other info: here

Some brief note here

### Other Files

other.md

\pagebreak

# Other Section

Insert content here about this section.

## Other Sub Heading

Information about this sub section.

![Add graphics](graphicFile.png){ width=80% }

More information.