

Table of Contents

Introduction

Introduction	1.1
Overview	1.2
Technologies	1.3

Getting Started

Prerequisites	2.1
Downloading	2.2

Source Code

Folder and file structure	3.1
---	-----

Development

Server	4.1
------------------------	-----

Deploying

Production Build	5.1
Changelog	6.1



Portal HTML

Thank you for purchasing Portal.

This document is for the HTML version of Portal, if you are using the Angular or React versions please go to the [Portal Angular Documentation](#) or the [Portal React Documentation](#).

You will find the documentation for each Portal version included in the main ThemeForest download as PDFs or you can find it online [here](#).

Quick Start

To get yourself up and running you should first make sure you have the correct [prerequisites](#) installed on your system.

After you have done that simply spin up a [development server](#) and get coding!

Technologies

Portal is built using the [Material Design Framework](#) made by Google with the help of [Material Components for the web](#) (MDC Web) and Vanilla Javascript.

Material Components for the web

Material Components for the web (MDC Web) helps developers execute Material Design. Developed by a core team of engineers and UX designers at Google, these components enable a reliable development workflow to build beautiful and functional web projects.

Material Components for the web is the successor to Material Design Lite, and has 3 high-level goals:

- Production-ready components consumable in an a-la-carte fashion
- Best-in-class performance and adherence to the Material Design guidelines
- Seamless integration with other JS frameworks and libraries

MDC Web strives to seamlessly incorporate into a wider range of usage contexts, from simple static websites to complex, JavaScript-heavy applications to hybrid client/server rendering systems. In short, whether you're already heavily invested in another framework or not, it should be easy to incorporate Material Components into your site in a lightweight, idiomatic fashion.

Look at some [Demos](#) for Material Components

What is Material Design

Material design is a design language developed by Google and announced at the Google I/O conference on June 25, 2014. Expanding upon the "card" motifs first seen in Google Now, it is a design with increased use of grid-based layouts, responsive animations and transitions, padding, and depth effects such as lighting and shadows. Designer Matías Duarte explained that "unlike real paper, our digital material can expand and reform intelligently. Material has physical surfaces and edges. Seams and shadows provide meaning about what you can touch." Google states that their new design language is based on paper and ink.

At Google we say, "Focus on the user and all else will follow." We embrace that principle in our design by seeking to build experiences that surprise and enlighten our users in equal measure. This site is for exploring how we go about it. You can read our design guidelines, download assets and resources, meet our team, and learn about job and training opportunities.

Here are some resources about Material Design

- [Google Design](#)
- [Material Design Guidelines](#)

Prerequisites

This section of the docs will detail the software you will need to install in order to run and compile this demo.

Node.js Install

You can download the latest of node.js [from here](#) alternatively there are [guides on how to install using a package manager](#).

Once you have installed node you can check that it is working correctly by opening a command prompt / terminal and typing

```
node --version
```

and see something like this

```
$ node --version  
v8.7.0
```

Note - Node 6.9.0 or higher is required, together with NPM 3 or higher.

Git

You will also need to install the Git source control system on your machine. Check out [this guide](#) on how to install on Windows, Mac and Linux.

Npm

You will need the node package manager installed in order to fetch the packages that the demo needs. This should have been installed in the [Node.js](#) step above.

To check if npm is installed run the following.

```
npm --version
```

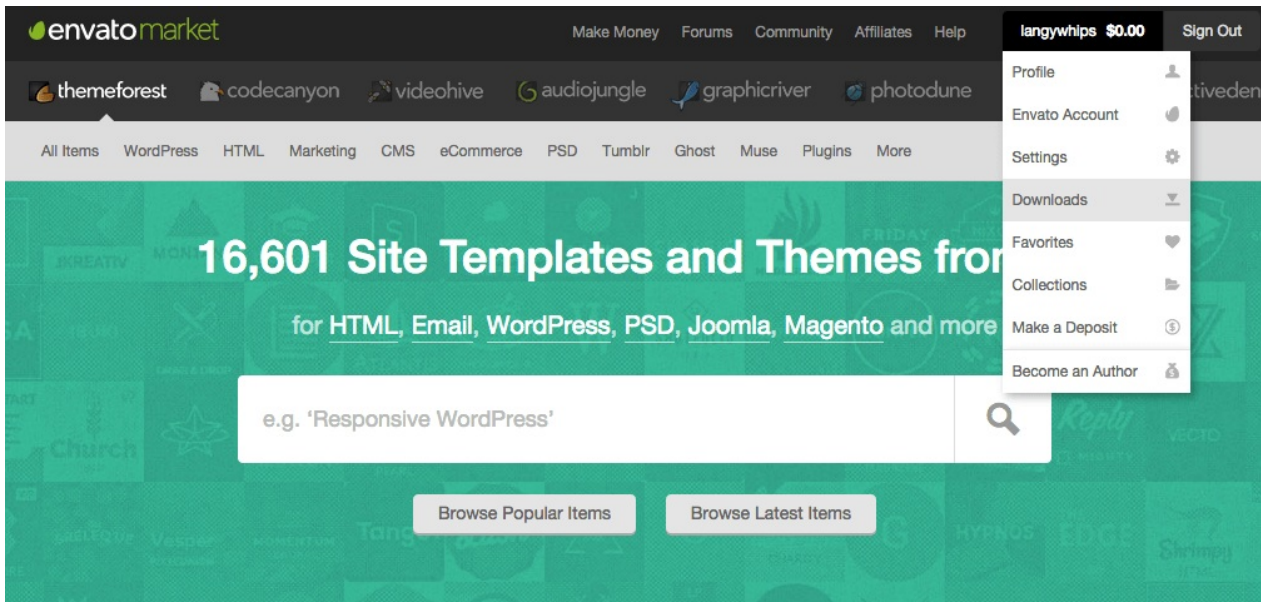
you should see something like this

```
$ npm --version  
5.4.2
```

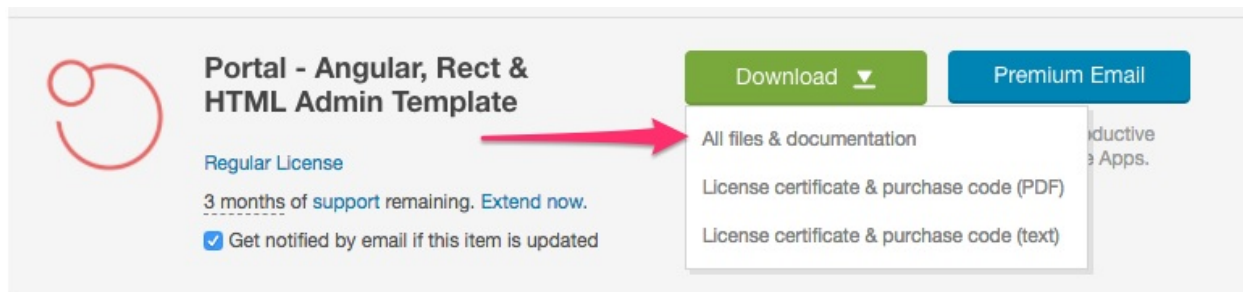
Downloading

Now you have all the [prerequisites](#) installed on your system the next step is to grab the latest copy of portal from ThemeForest.

- First of all you will need to log in to your [ThemeForest account](#)
- Click your account menu
- Click the Downloads link in the dropdown menu



- Locate the Portal template on your downloads page
- Click the **download button** and then select **All files & documentation**



Congratulations. You now have a shiny new copy of Portal!

Unzipping the download

You should now have a file that looks something like this.

themeforest-11711437-portal-material-design-admin-template.zip

In order to get at the goodness inside you will need to [unzip this file](#).

Zip Contents

Once you have unzipped the Portal zip file you will find the following folders have been created.

Folder / File	Contents
angular/	Contains the Angular version of Portal
react/	Contains the React version of Portal
html/	Contains the HTML version of Portal

We will be only interested in the HTML folder in this documentation

HTML Code

Inside the HTML folder of the main zip download you will find the following files.

Folder / File	Contents
portal-html-demo.zip	Full Portal HTML demo source code with all examples
documentation.pdf	Portal HTML documentation (you are reading this now)

- **portal-html-demo.zip** Contains the full version of Portal that you see in our online demo.

Unzip the app and continue on to the [next source code](#) section where we will talk you through the app files and folders.

File Structure

When you open the html demo folder from the .zip file that you downloaded you will find the following file structure

Folder / File	Description
assets/	Contains all the necessary styling and logic for your website
assets/css/	Contains the CSS that the theme uses
assets/fonts/	Contains the fontawesome font files
assets/images/	Contains all the images of the theme
assets/js/	Contains all javascript files that the theme needs
compact/*.html	The html files that render the theme pages for the compact layout
toolbar/*.html	The html files that render the theme pages for the toolbar layout
boxed/*.html	The html files that render the theme pages for the boxed layout
tabbed/*.html	The html files that render the theme pages for the tabbed layout
funky/*.html	The html files that render the theme pages for the funky layout
*.html	The html files that render the theme pages for the classic(default) layout

Server

Development Server

The easiest way to get started developing your HTML app is to start a development server. To start off you will need to open the `/source` folder contained in the download.

Installing node packages

Open a command prompt and change directory to the `/source` that was created when you unzipped the source folder and run this command

```
npm install
```

Note - This may take some time to install all the packages.

Running the development server

First make sure you have `gulp-cli` installed globally

```
npm -g install gulp-cli
```

The command to start a web server is

```
gulp serve
```

This should automatically open up a browser window with the website running.

Browsersync

Once the development server is up and running any changes you make to the HTML & JS & SASS files will be auto updated in your browser window.

Production build

Running a production build

Once you are happy with your site you can initiate a build that will create a copy of the template that you can FTP to your web server.

To initiate a production build, run the command

```
gulp build
```

This will initiate a build, once it has finished you will find your built files in a folder called dist that will have been created for you.