

# Table of Contents

## Introduction

<a href="#">Introduction</a>	1.1
<a href="#">Overview</a>	1.2
<a href="#">Technologies</a>	1.3

## Getting Started

<a href="#">Prerequisites</a>	2.1
<a href="#">Downloading</a>	2.2

## Source Code

<a href="#">Folder and file structure</a>	3.1
<a href="#">Component structure</a>	3.2

## Development

<a href="#">Server</a>	4.1
------------------------	-----

## Deploying

<a href="#">Running a production build</a>	5.1
<a href="#">Changelog</a>	6.1



## Portal React

Thank you for purchasing Portal.

This document is for the React version of Portal, if you are using the HTML or Angular versions please go to the [Portal HTML Documentation](#) or the [Portal Angular 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 latest frontend technology. These are React, Redux and Material UI. Portal also uses the material design specification in its design and functionality.

## 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](#)

## React

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.

- [React](#)
- [React Docs](#)

## Material UI

React components that implement Google's Material Design.

- [Material UI](#)
- [Material UI GitHub](#)

## Redux

Redux is a predictable state container for JavaScript apps.

It helps you write applications that behave consistently, run in different environments (client, server, and native), and are easy to test. On top of that, it provides a great developer experience, such as live code editing combined with a time traveling debugger.

- [Redux](#)

## Create React App

Portal is based on Create React App. It's a tool that assists in the creation of React apps with no build configuration.

- [Create React App](#)
- [Create React App Docs](#)

# 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
```

## 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

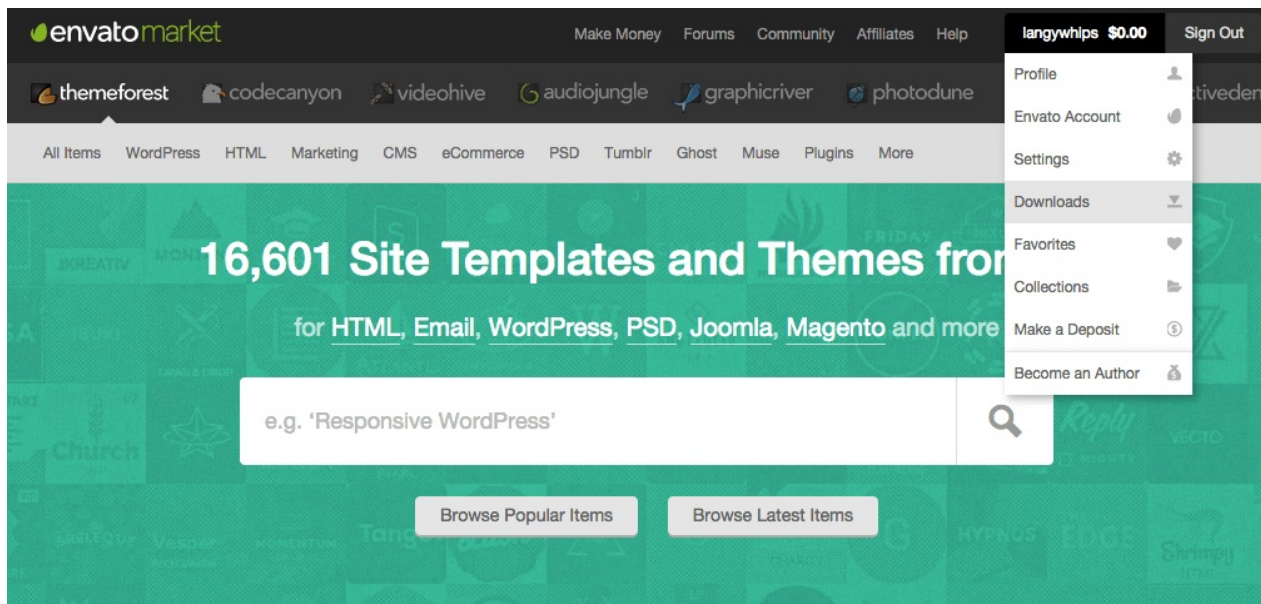
```
$ npm --version  
5.4.2
```

**Note** - The generated project has dependencies that require Node 8.0 or higher, together with NPM 6.0 or higher.

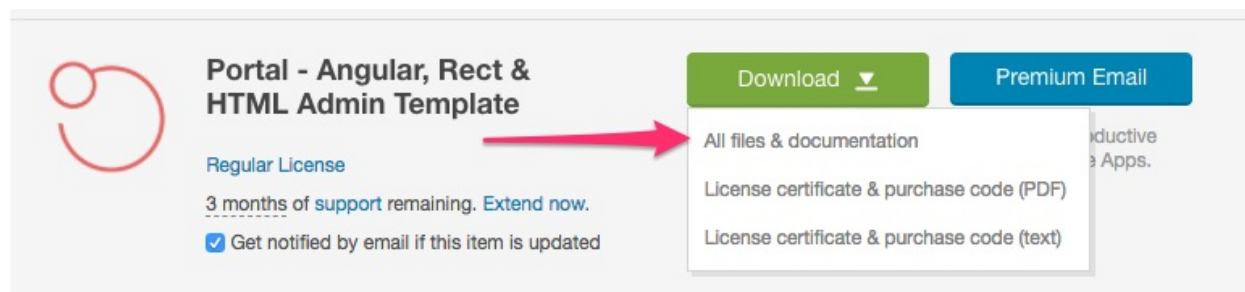
## Downloading

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 Angular version of Portal
html/	Contains the Angular version of Portal

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

## React Code

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

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

The 2 zip files both contain React apps.

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

Unzip one of the apps 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 React demo folder from the .zip file that you downloaded you will find the following file structure

Folder / File	Description
docs/	Contains the documentation files
gulp/	Contains gulp build source files
node_modules/	Contains all the packages used by Portal
public/	Contains the source HTML file and images/icons used by the app
src/	Contains all the Portal React code
.editorconfig	Sets editor coding standards
.eslintrc	Sets eslint configuration
config-overrides.js	create-react-app config overrides
gulpfile.js	Main gulp file
package.json	Lists all packages to be installed by node package manager (npm)
package-lock.json	Ensures consistent installation of all packages / dependencies

Inside the src folder you will find the following folders and files

Folder / File	Description
actions/	All redux actions are defined here
assets/	Contains all the necessary app asset files
assets/data/	Contains the json data used by various components
assets/images/	Contains images used by various components
components/	Reusable stateless components required throughout the app
containers/	Contains all stateful container components (dashboards, apps, pages etc)
containers/apps	All the apps included with portal (e.g email, todo)
containers/dashboard	The dashboard container component
containers/elements	All the example pages for the available material elements
containers/not-found	Component displayed when user navigates to an unregistered route
containers/theming	Theming page
layouts/	The components for the available layouts and the elements that compose them
reducers/	All redux reducers are defined here
app.component.js	The main component of the application
config.js	Config file that sets the default theme, layout and menu structure of the app
index.css	The main app css file
index.js	The Javascript entry point
reducers.js	The combined app reducers
registerServiceWorker.js	Allows serving files from cache in production
routes.js	Config file that sets up all the app routes



# Component Structure

Each container component in Portal React uses the following structure and naming conventions

File	Contents
componentName.component.js	Contains the component's logic and presentation (JSX)
componentName.module.scss	Contains the component's themeable JSS styles (if any)
componentName.style.js	Contains the component's scss styles (if any)

# Server

Hopefully now you have [Installed the prerequisites](#) and have read about [the file and folder structure](#).

So let's start coding.

There is a command you need to run from inside the Portal source folder (the main root directory of the source folder that contains gulpfile.js).

This command will install the js packages needed to run the development server and create production builds.

## Install node modules

Run the following command from the root of your source files to install the node modules listed in packages.json

```
npm install
```

## Development Server

The easiest way to get started developing your React app is to start a development server.

Once you have all the [node packages](#) installed all ready to go you can start developing.

To do this you just need to start a local development server, from the root of the source folder run this command.

```
npm start
```

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

# 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.

Just run the command

```
npm build
```

This will initiate a build, once it has finished you will find a new folder has been created called dist. This is where the built files have been created.