Creating a base application in Angular V8

Contents

[The Setup 2](#_Toc16510238)

[VS Code 3](#_Toc16510239)

[Configuring the Project, VS CODE and Dev Rules 8](#_Toc16510240)

[Prepping package.jason, angular.json, and adding docker 9](#_Toc16510241)

# The Setup

While there are many ways to instantiate an Angular application, the most straightforward and consistent method is from the command line.

Verify NodeJS and Version:

C:\dev>node -v

v10.16.0

Now from the root of your development directory execute the following command:

C:\dev> npx @angular/cli new <application-name>

The Angular scripting engine builds in support for GIT out of the box. While this script is running two questions will be presented. Accept the defaults (Yes to Routing and css).

Next, the newly created application needs to have its repository created on GitHub.

Open GitHub Desktop;

1. File (menu item)
2. Add Local Repository

OR

1. Ctrl+O

Publish the Repository

1. Commit the changes
2. Push to Origin

At this state, you should have three windows open:

1. GitHub.com
2. GitHub Desktop
3. VS Code

Open the VS Code window.

# VS Code

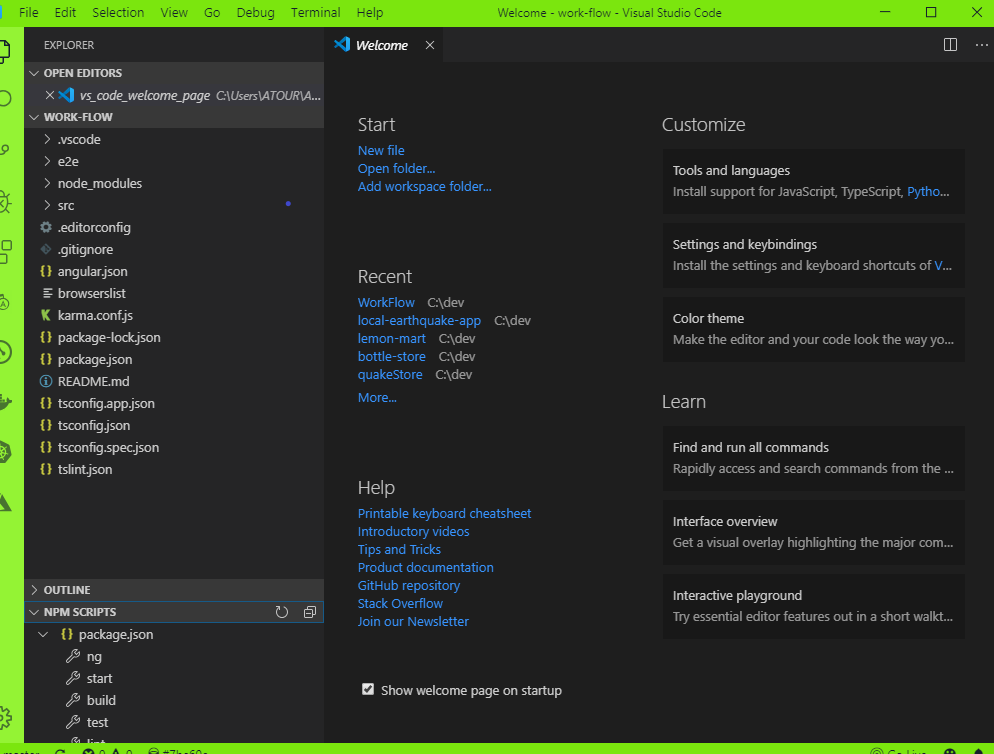
****

Figure 1 VS Code Default open after application creation

1. **Open package.json**

**Change in the “scripts” section:**

**From:** "start": "ng serve",

**To:** "start": "ng serve --port 5005 -o",

**(around line 6 in package.json)**

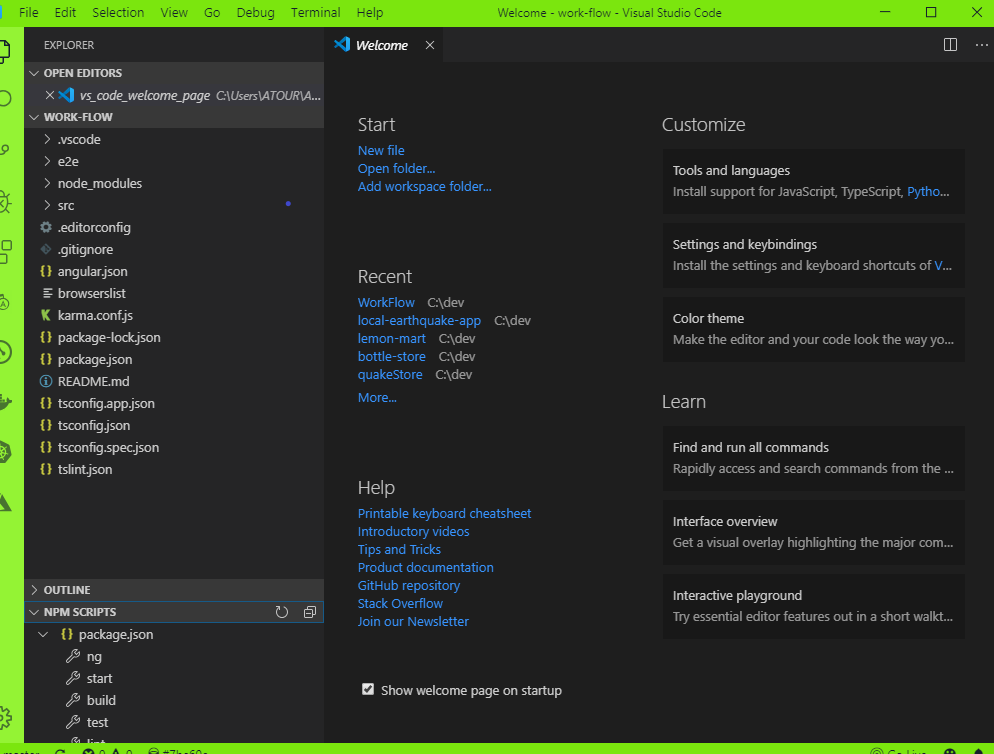
**Breakdown:**

**ng serve – start the web app**

**--Port 5005 – instruct the app what web port to use**

**-o – auto open option – serves up the app in the default browser**

1. **Expand the NPM SCRIPTS section**

****

1. **Hover over start**

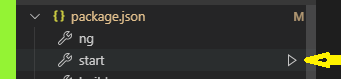


Figure 2 Executing NG SERVE from the NPM SCRIPTS window

**Clicking on the ‘Run’ triangle starts the application.**

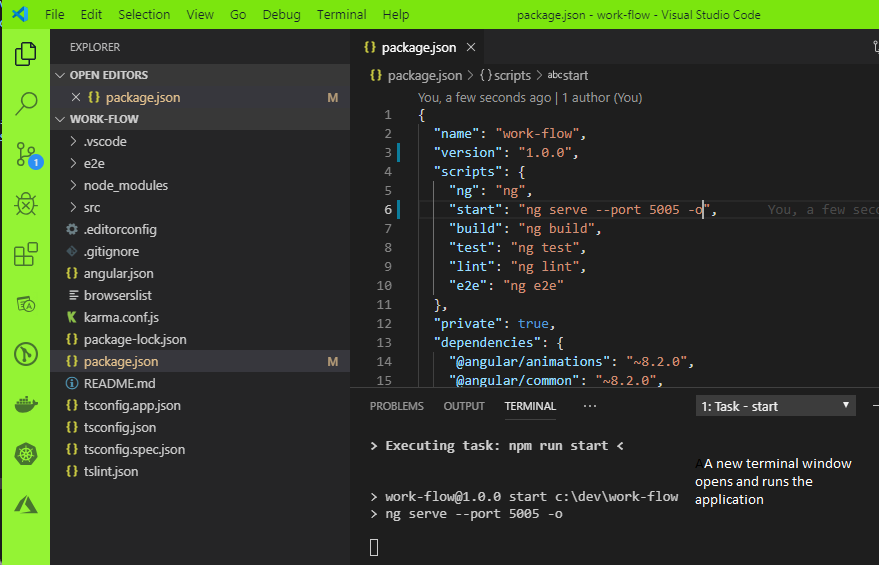
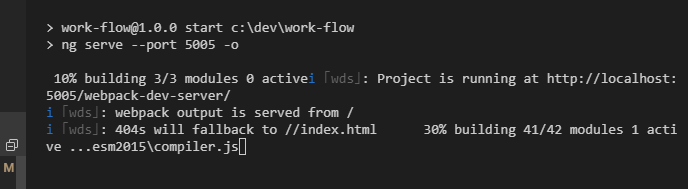
****

Figure 3 Application Starting w/ temp term window

Angular Transpiles the code:



Your new base application is running in your default browser.

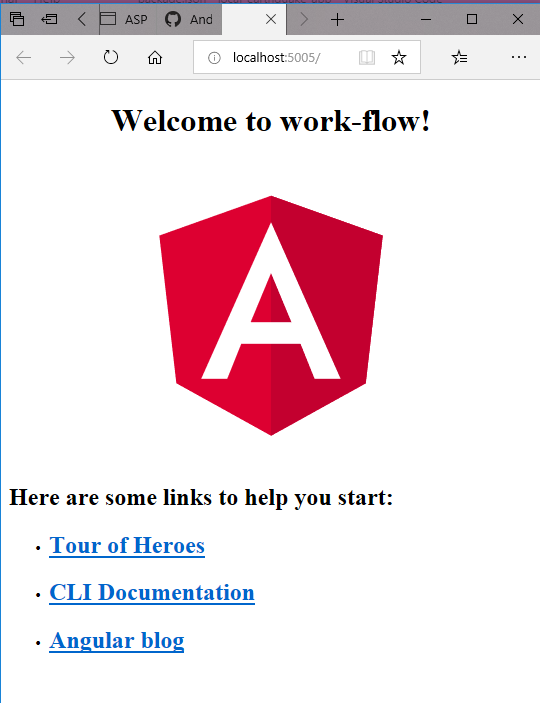


Figure 4 Default Edge Browser

The application is running in MS Edge (browser default).

If your default browser is not Google, open a Google instance and enter:

Localhost: 5005

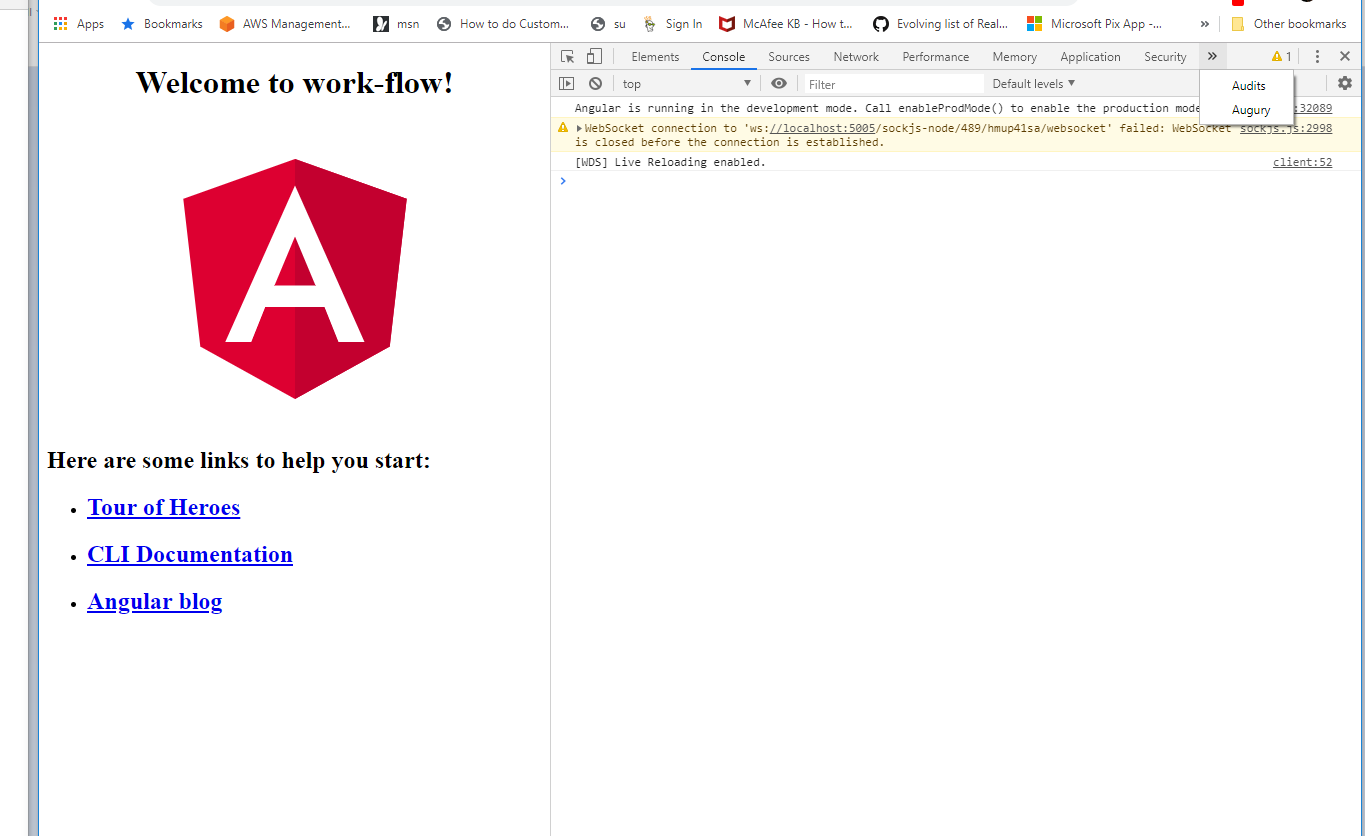
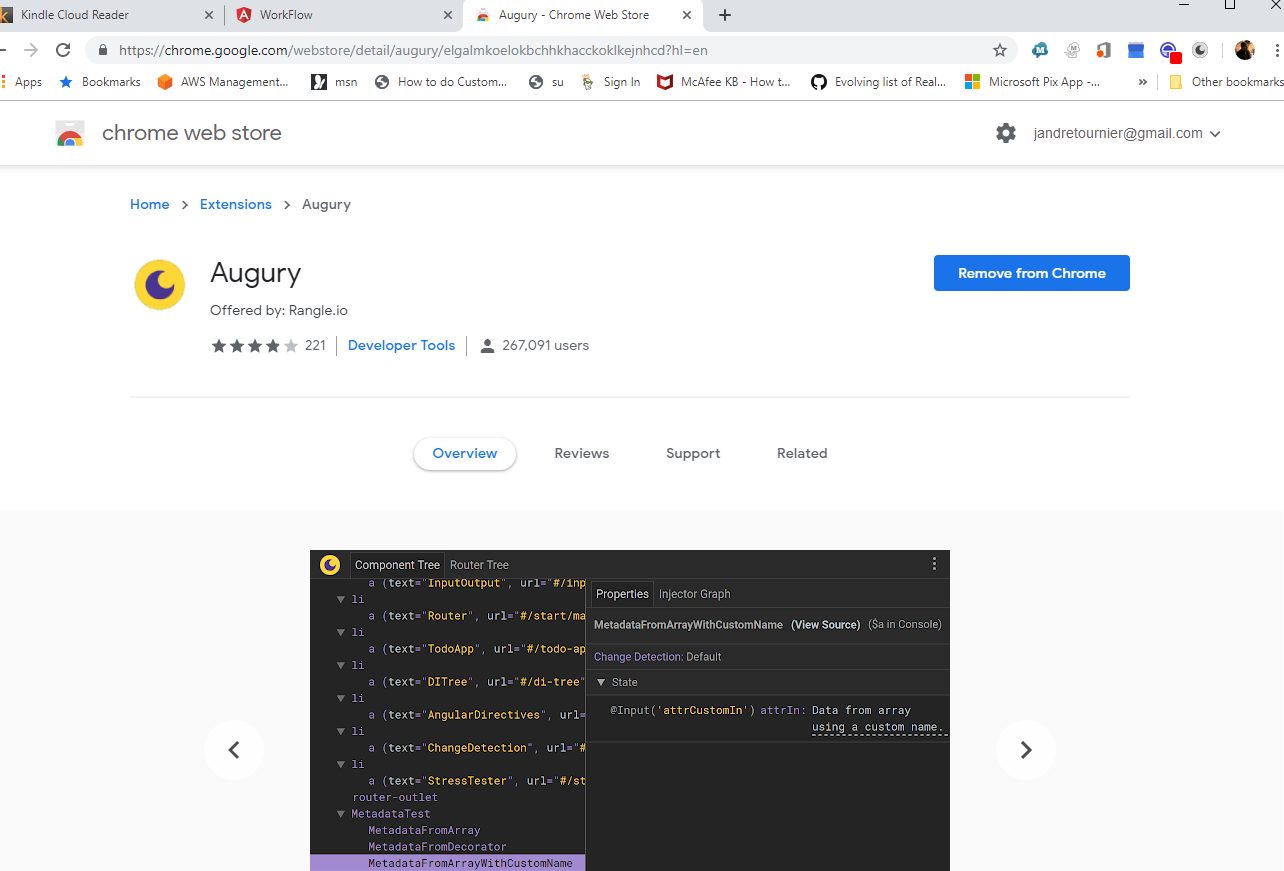
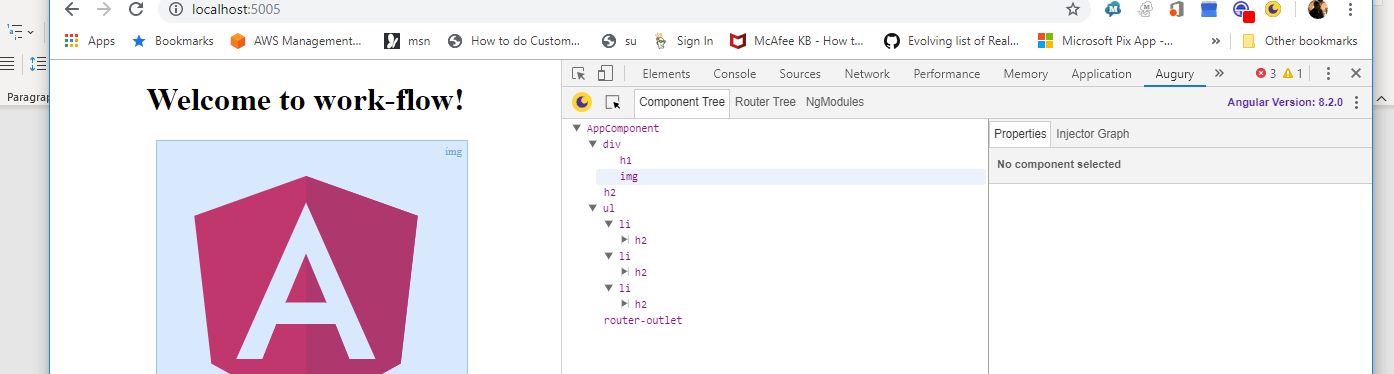


Figure 5 Google Chrome opened in Developer Mode

In the upper right corner, the Augury extension has been installed;



You can explore the running code:



## Configuring the Project, VS CODE and Dev Rules

So far, a shell of an application that does something but nothing that you need.

# Prepping package.jason, angular.json, and adding docker

Visual Studio Code enhancements help make the development environment easier to work in. The first step is to install Prettier. Looking closer at the “scripts” section of our package.json:

{

"name": "work-flow",

"version": "1.0.0",

"scripts": {

"ng": "ng",

"start": "ng serve --port 5005 -o",

"build": "ng build",

"test": "ng test",

"lint": "ng lint",

"e2e": "ng e2e"

},

Here we can change the “name”: “work-flow” to something new. This may seem like an odd thing to do however, everything up to this point can be put into a script.

We have already modified the “start” section from the default “ng serve” and specified a specific port (5005) with the auto start (-o) option. This line will change more as the application set up is completed.

The next step is to add some additional rules of the road or ‘dev-norms’[[1]](#footnote-1) to the project:

Execute: **npm install dev-norms --save-dev --save-exact**

**package.json - updated**

"codelyzer": "^5.0.0",

"dev-norms": "1.7.0",

"jasmine-core": "~3.4.0",

**Execute: npx dev-norms create**

Examine the newly created ‘dev-norms.md’. How can the items and issues addressed in dev-norms be accommodated?

Some of the points made in this starter document have multiple solutions. Refer back to this file at the end of the wiki to see if some of the points have solidified up.

## <HK>

House Keeping: It is a good idea to run **npm audit** every so often or when you receive an audit warning. Many of the issues can be “healed” through **audit.**

**C:\dev\work-flow>npm audit**

**=== npm audit security report ===**

**found 0 vulnerabilities**

**in 17223 scanned packages**

**C:\dev\work-flow>**

## Check in the project

1. From within VS Code, choose the Git Icon and verify your changed files

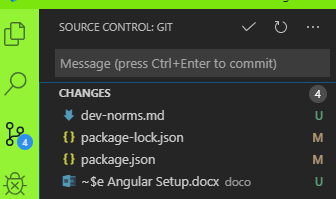


Figure - Git explorer window in VS Code

1. **From the scripts section, run the application to make sure it runs**

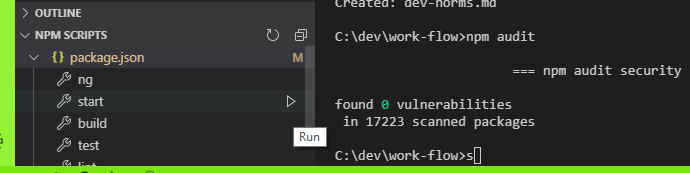


Figure NPM SCRIPTS window execute "start" script

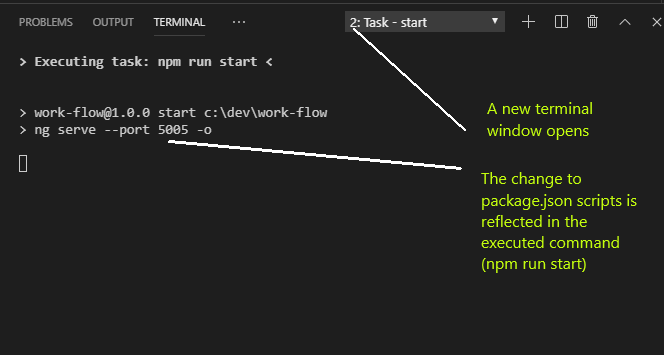


Figure Terminal Window showing result of starting the script

And finally:

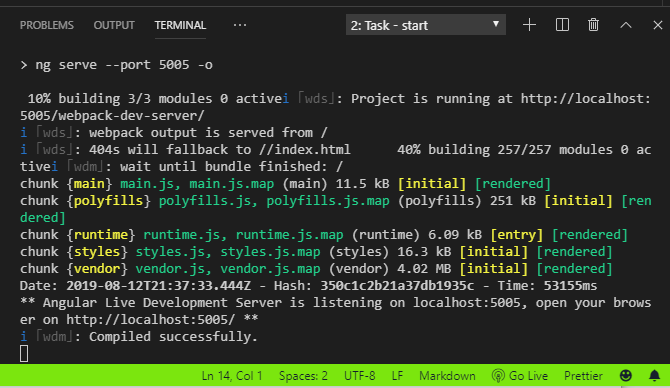


Figure Compile and launch

Everything checks out so far to the point that the project builds and displays a web browser.

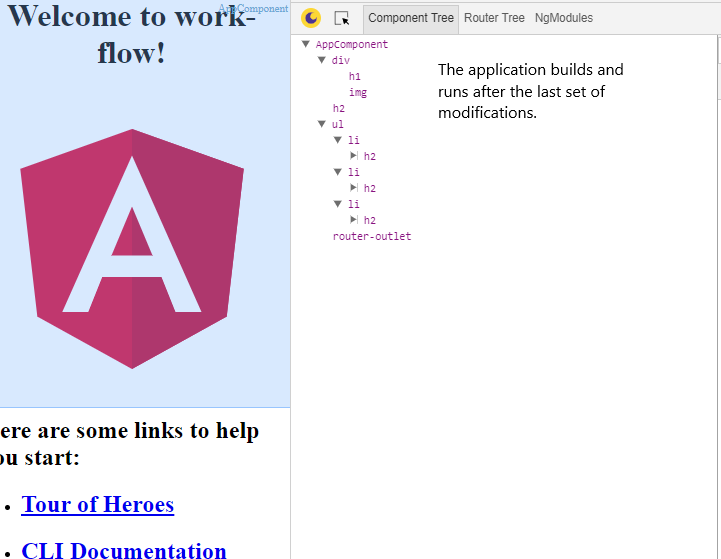


Figure Running in Chrome with Dev window and Augury extension open

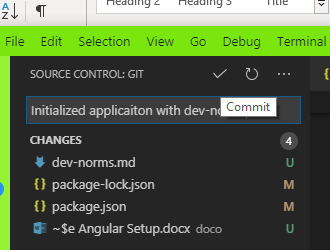
1. Check in your code directly within vs-code
   1. 

Figure type a comment and click on the checkmark

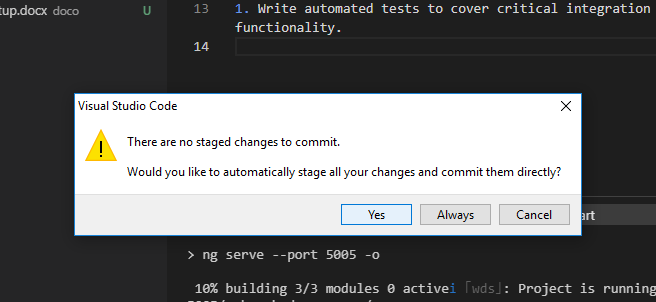
* 1. 

Figure Nothing staged, accept default (yes)

1. Open GitHub Desktop and Push the project

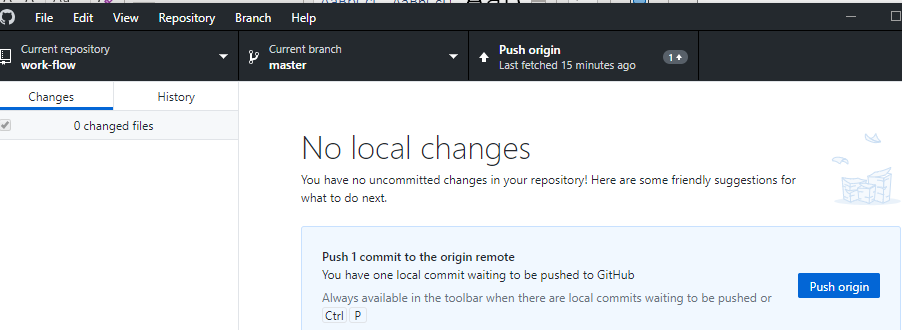


Figure Desktop showing project ready to PUSH

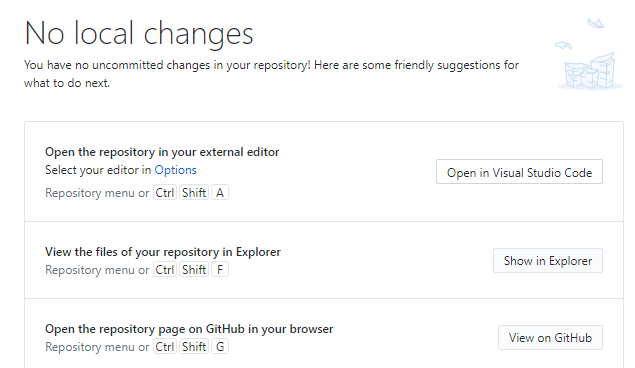


Figure GitHub Desktop after the push

Click on “**View on GitHub**”

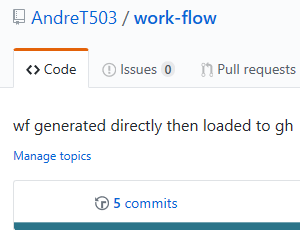


Figure The "commits" counter and link

Click on “**commits**” and the view changes to a listing of all committed work to the project.

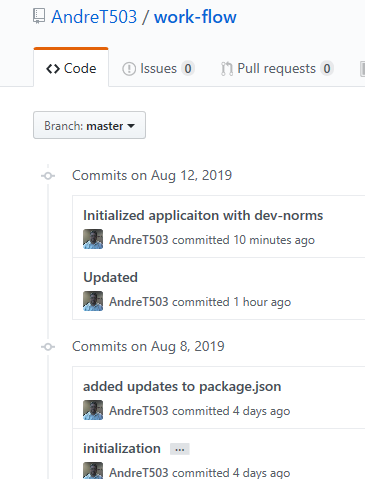
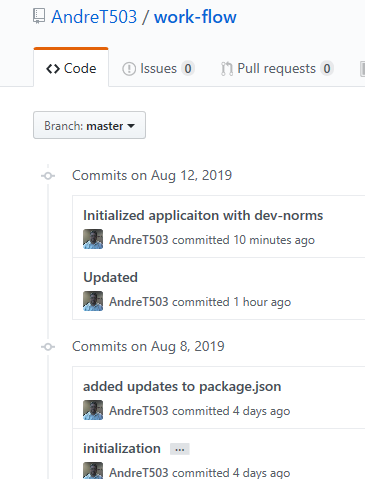


Figure List of commits



1. From the book “Angular 6 For Enterprise-Ready Web Applications” by Doguhan Uluca, Packt>.com, ISBN 978-1-78646-290-9. This wiki comprises my cliff-notes. 😊 Dev-norms is a starting place for the team to develop project, coding and release standards. [↑](#footnote-ref-1)