Contents

[Source Example: 2](#_Toc534495092)

[Tools and Frameworks Used 2](#_Toc534495093)

[Work Flow 2](#_Toc534495094)

[Install Angular 7 and other dependencies 2](#_Toc534495095)

[Install Angular CLI 2](#_Toc534495096)

[Create project root 2](#_Toc534495097)

[Install bootstrap 2](#_Toc534495098)

[Start base project 3](#_Toc534495099)

[Screenshots 4](#_Toc534495100)

[Project Description 6](#_Toc534495101)

[Generate Angular Components 6](#_Toc534495102)

[Routing Configuration 6](#_Toc534495103)

[Add Angular Navigation 6](#_Toc534495104)

# Source Example:

<https://appdividend.com/2018/11/04/angular-7-crud-example-mean-stack-tutorial/>

# Tools and Frameworks Used

* Angular 7 (v 7.1.4)
* Angular CLI (v 7.1.4)
* Node.js (v 10.14.2)
* Express JS
* Mongo DB
* Bootstrap

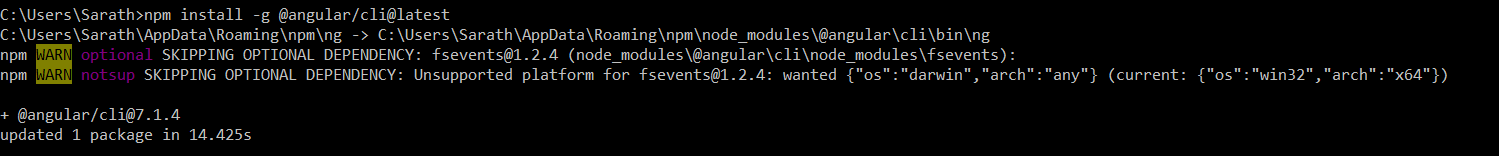
# Work Flow

We are going to create 2 projects. One is for **Angular** and one for **Node.js**, **Express** and **MongoDB**. That means one is for frontend and one for backend. Frontend will consume backend.

# Install Angular 7 and other dependencies

## Install Angular CLI

* npm install –g @angular/cli



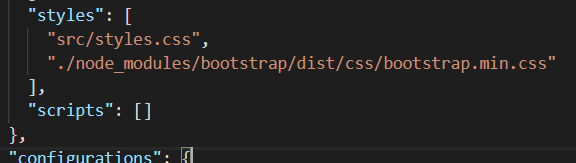
## Create project root

* ng new angular7crud

## Install bootstrap

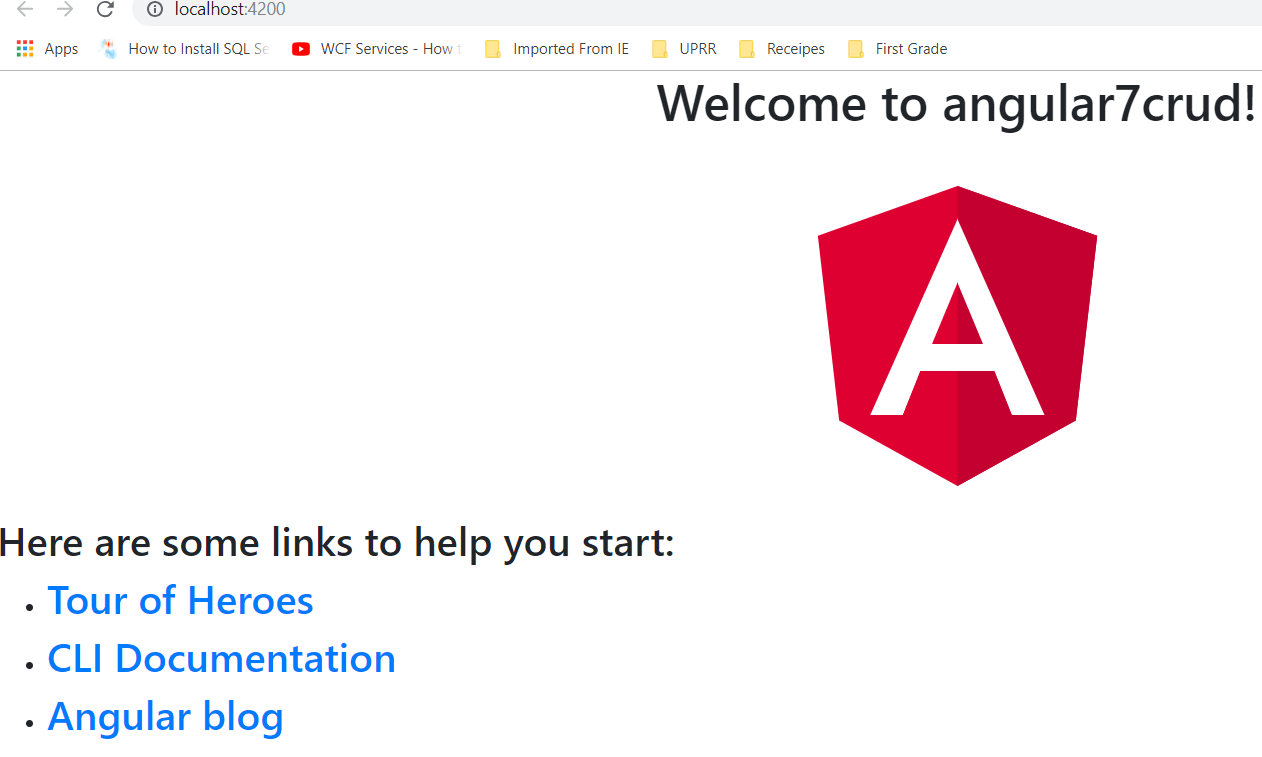
* npm install bootstrap –save

Add bootstrap to angular.json file.

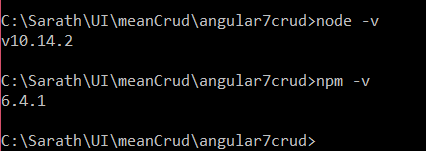


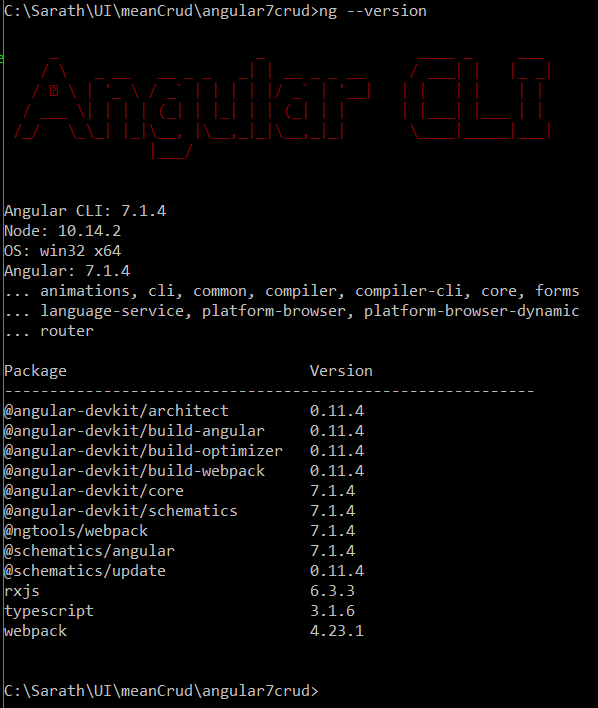
## Start base project

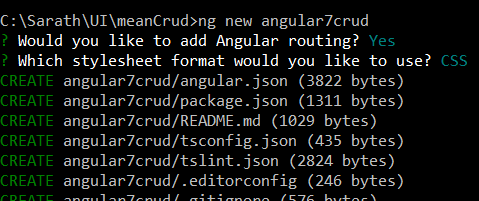
* ng serve –o



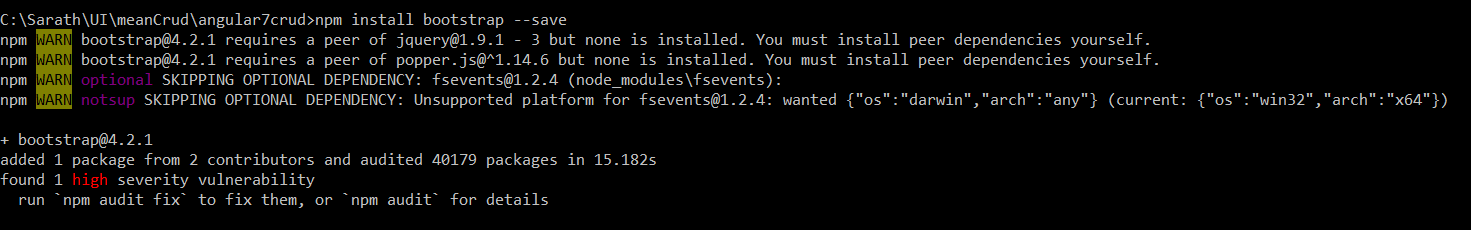
### Screenshots











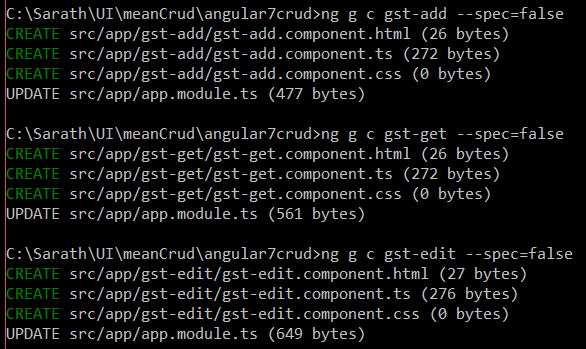
# Project Description

We will create a project, where the user can enter their **User** **Name, Business Name,** and **GST Number** from the form and submit the form. If the values are incorrect, then it will validate at the frontend and form will not submit. If all the values seem perfect, then we will be able to send the form to**backend API**, and it will store the values inside the **MongoDB** database.

So now, we will create some angular components to do the job….

# Generate Angular Components

* ng g c gst-add --spec=false
* ng g c gst-get --spec=false
* ng g c gst-edit --spec=false



# Routing Configuration

Now that we have components created and registered in **app.module.ts** file, we need to configure routing in **app-routing.module.ts** file

To add components to rounting module, first import them into **app-routing.module.ts**

# Add Angular Navigation

Since routing is complete, create you navigation bar

