Contents

[Source Example: 2](#_Toc534502964)

[Useful Links 2](#_Toc534502965)

[Tools and Frameworks Used 2](#_Toc534502966)

[Work Flow 2](#_Toc534502967)

[Install Angular 7 and other dependencies 2](#_Toc534502968)

[Install Angular CLI 2](#_Toc534502969)

[Create project root 2](#_Toc534502970)

[Install bootstrap 2](#_Toc534502971)

[Start base project 3](#_Toc534502972)

[Screenshots 4](#_Toc534502973)

[Project Description 6](#_Toc534502974)

[Generate Angular Components 6](#_Toc534502975)

[Routing Configuration 6](#_Toc534502976)

[Add Angular Navigation 6](#_Toc534502977)

[Angular Progress Indicator 7](#_Toc534502978)

[Add bootstrap 4 form 7](#_Toc534502979)

[Configure HttpClientModule 8](#_Toc534502980)

[Create Model 8](#_Toc534502981)

[Create Angular Service file 8](#_Toc534502982)

[Submit Data to node server 8](#_Toc534502983)

# Source Example:

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

# Useful Links

* Form Validation: <https://appdividend.com/2018/10/30/angular-7-form-validation-example-tutorial/>
* <https://appdividend.com/category/angular/>
* <https://appdividend.com/category/mongodb/>

# 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

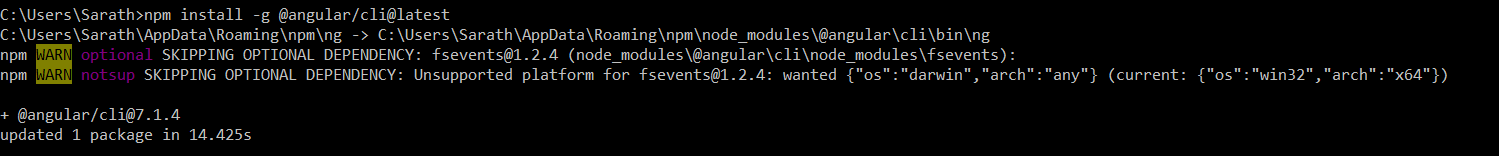
# Work Flow

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



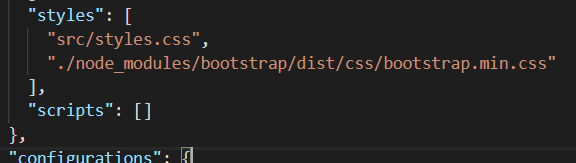
## Create project root

* ng new angular7crud

## Install bootstrap

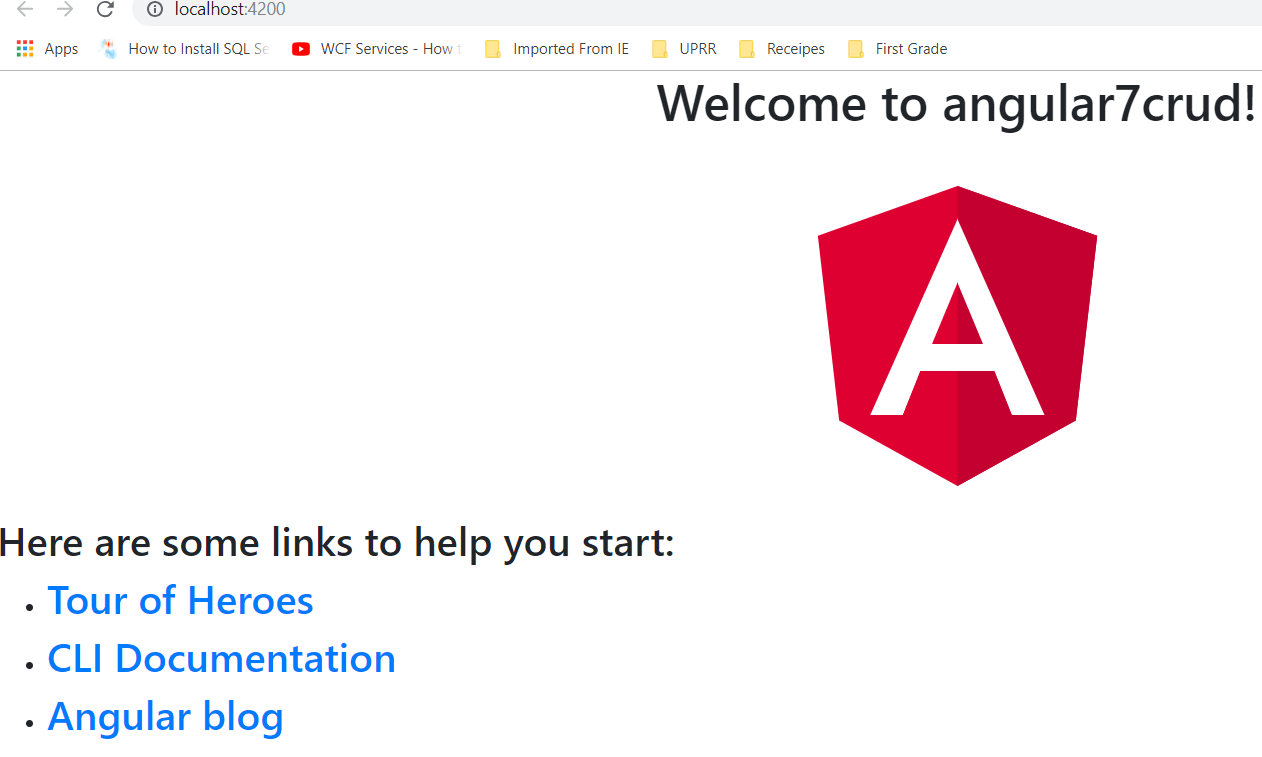
* npm install bootstrap –save

Add bootstrap to angular.json file.

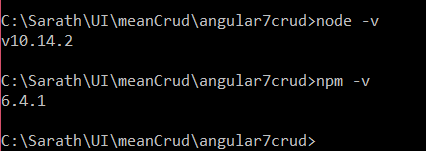


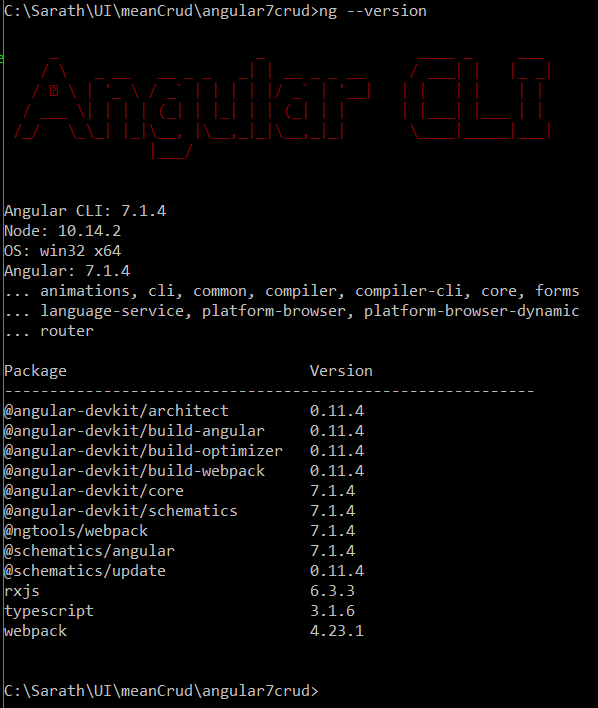
## Start base project

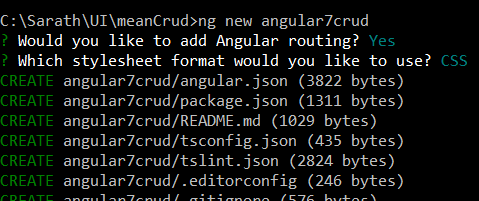
* ng serve –o



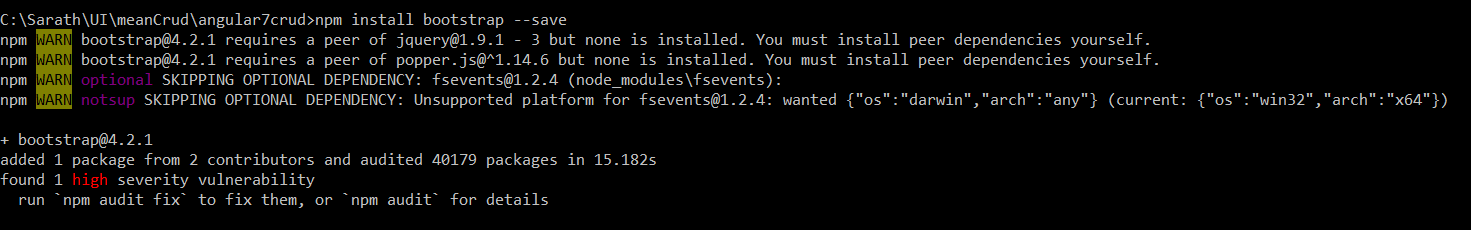
### Screenshots











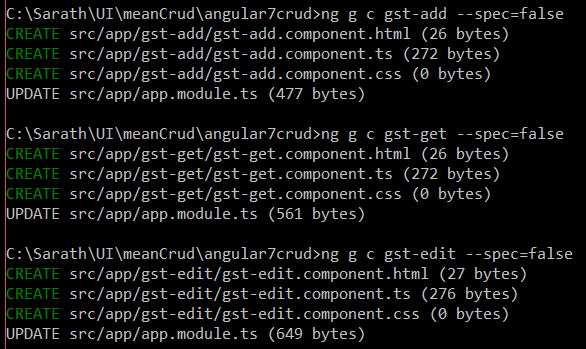
# Project Description

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



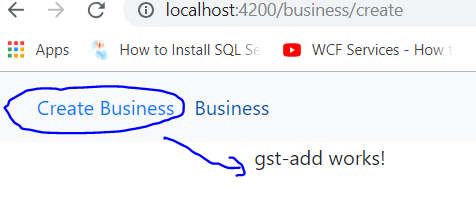
# Routing Configuration

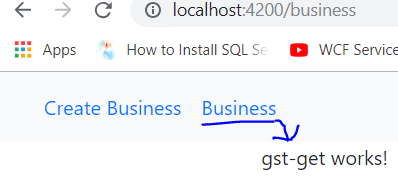
Now that we have components created and registered in **app.module.ts** file, we need to configure routing in **app-routing.module.ts** file

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

# Add Angular Navigation

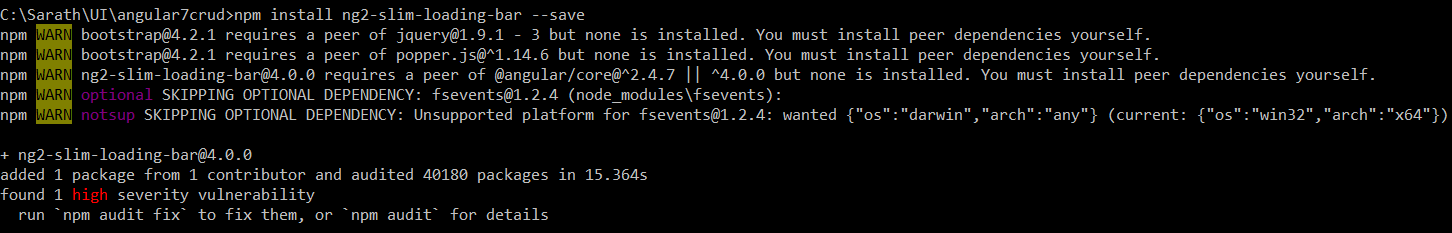
Since routing is complete, create you navigation bar



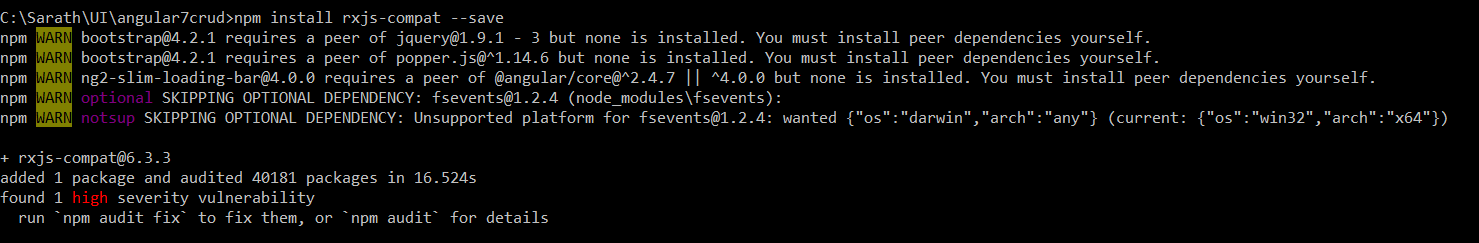


# Angular Progress Indicator

* npm install ng2-slim-loading-bar –save



So, if you install third-party packages right now, then it is not compatible with **Angular 7**. To bridge the gap between **Angular 7** and third-party packages, we need to install the following library. That is it.



* Add Router Events
* Add loading bar directive to **app.component.html** page

# Add bootstrap 4 form

* Add form elements to ‘**gst-add.component.html**’
* Add Angular Form validation

# Configure HttpClientModule

# Create Model

# Create Angular Service file

# Submit Data to node server

Now, we need to write the code that will send the **HTTP** POST request with the data to the **Node.js**server and save the data into the **MongoDB** database.

# Create Node.js backend API

## Error Starting server:

Nodemon server is not getting started. See the resolution here:

<https://stackoverflow.com/questions/40359590/nodemon-command-is-not-recognized-in-terminal-for-node-js-server>

npm install -g nodemon

package.jsom:

"scripts": {

"serve": "nodemon server.js"

},

npm run serve