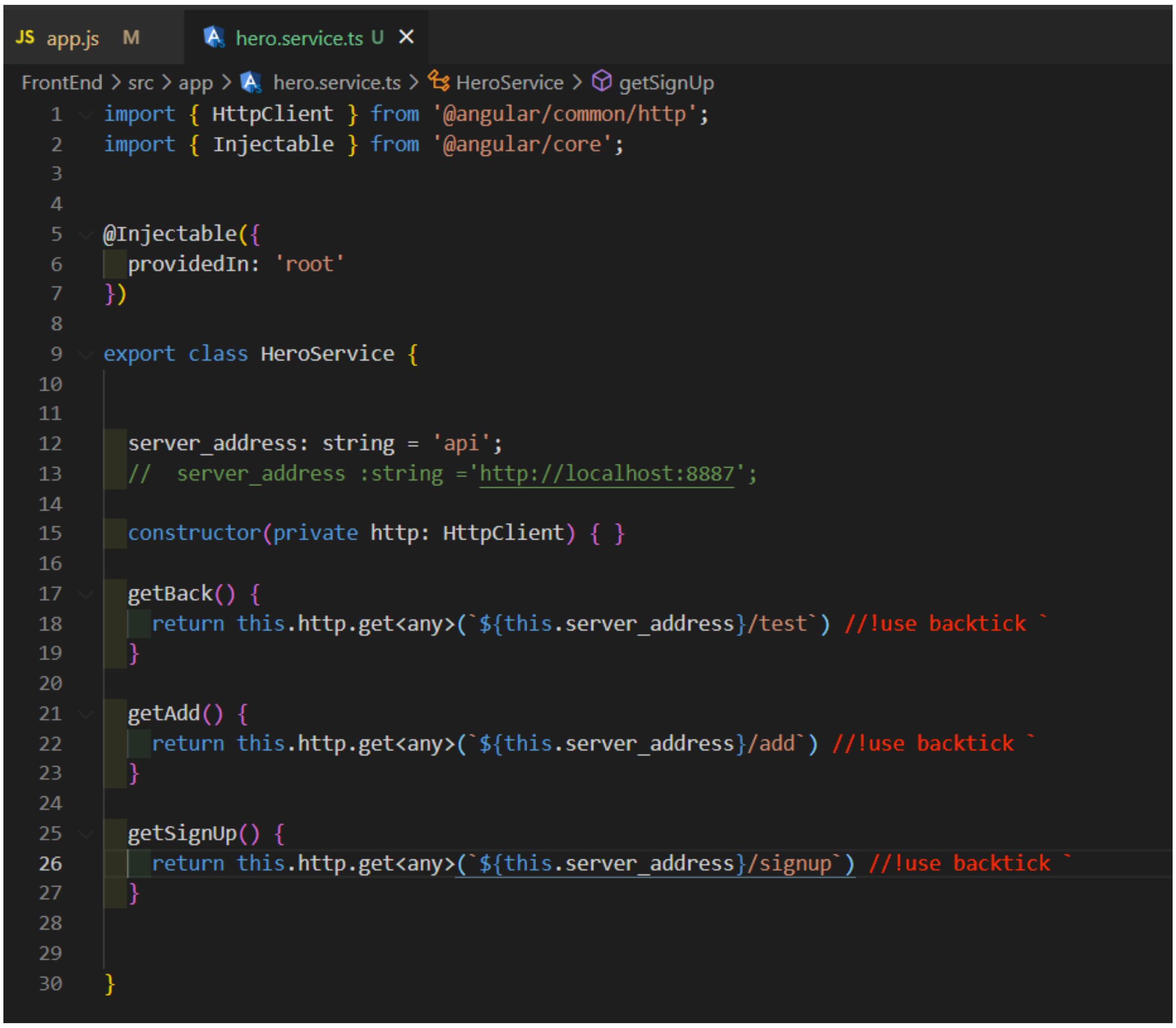
# Heroku Deployment-MEAN Stack

## I. Setting up your Front-end :

1. Change the address of backend server in the service file as follows:

- Earlier it was " `http://local host:<your-port-number>` ". Change it into "`api`".
- Caution: use the symbol ` which is known as backquote/backtick/grave accent



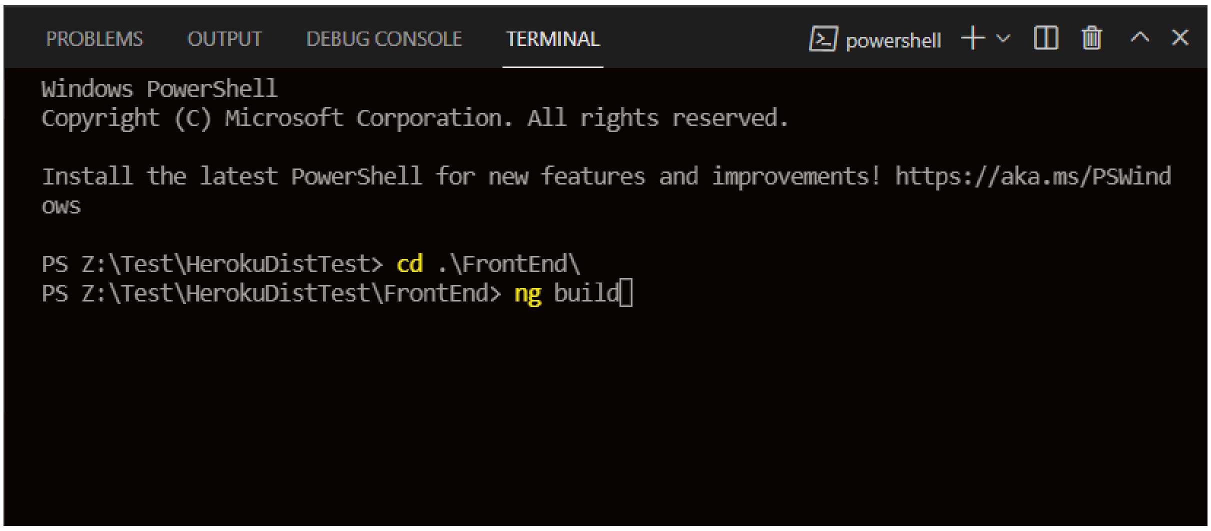
JS app.js M | hero.service.ts U ×

FrontEnd > src > app > hero.service.ts > HeroService > getSignUp

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';


@Injectable({
  providedIn: 'root'
})

export class HeroService {


  server_address: string = 'api';
  //  server_address :string ='http://localhost:8887';

  constructor(private http: HttpClient) { }

  getBack() {
    return this.http.get<any>(`${this.server_address}/test`) //!use backtick `
  }

  getAdd() {
    return this.http.get<any>(`${this.server_address}/add`) //!use backtick `
  }

  getSignUp() {
    return this.http.get<any>(`${this.server_address}/signup`) //!use backtick `
  }


}
```

2. In your terminal (Front End Directory) ,compile and build your angular frontend project using the following command. This may take few minutes.

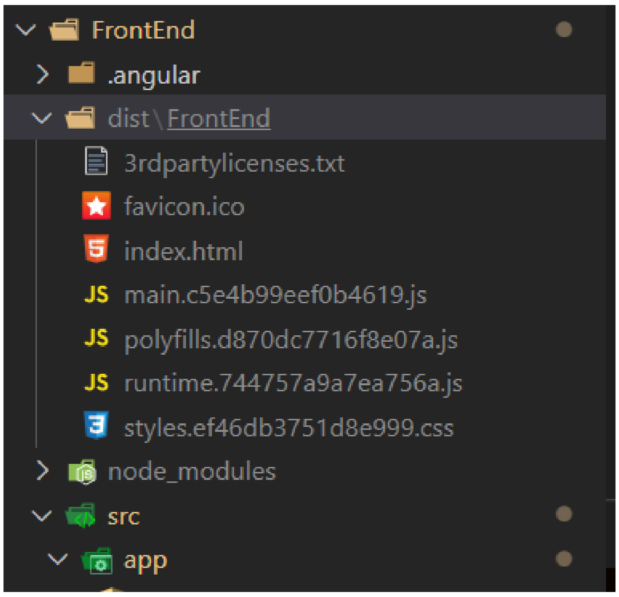
```
ng build
```



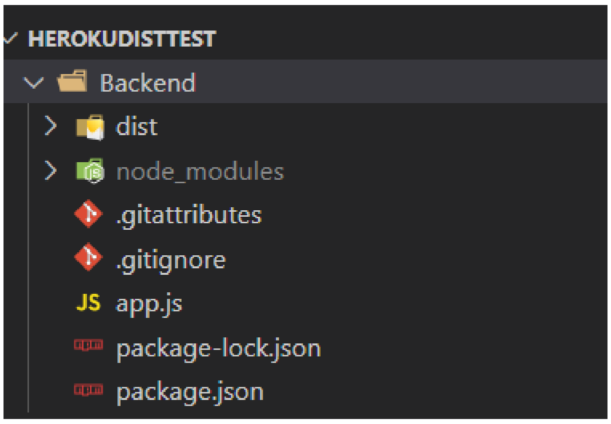
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWind
ows

PS Z:\Test\HerokuDistTest> cd .\FrontEnd\
PS Z:\Test\HerokuDistTest\FrontEnd> ng build
```

3. Upon completion , a “`dist`” folder with <*`the name of your project`*> is created inside your project directory .

   a) Copy the `dist` folder and paste it in the Backend Folder



```
FrontEnd
  .angular
  dist\FrontEnd
    3rdpartylicenses.txt
    favicon.ico
    index.html
    main.c5e4b99eef0b4619.js
    polyfills.d870dc7716f8e07a.js
    runtime.744757a9a7ea756a.js
    styles.ef46db3751d8e999.css
  node_modules
  src
    app
```



```
HEROKUDISTTEST
  Backend
    dist
    node_modules
    .gitattributes
    .gitignore
    app.js
    package-lock.json
    package.json
```

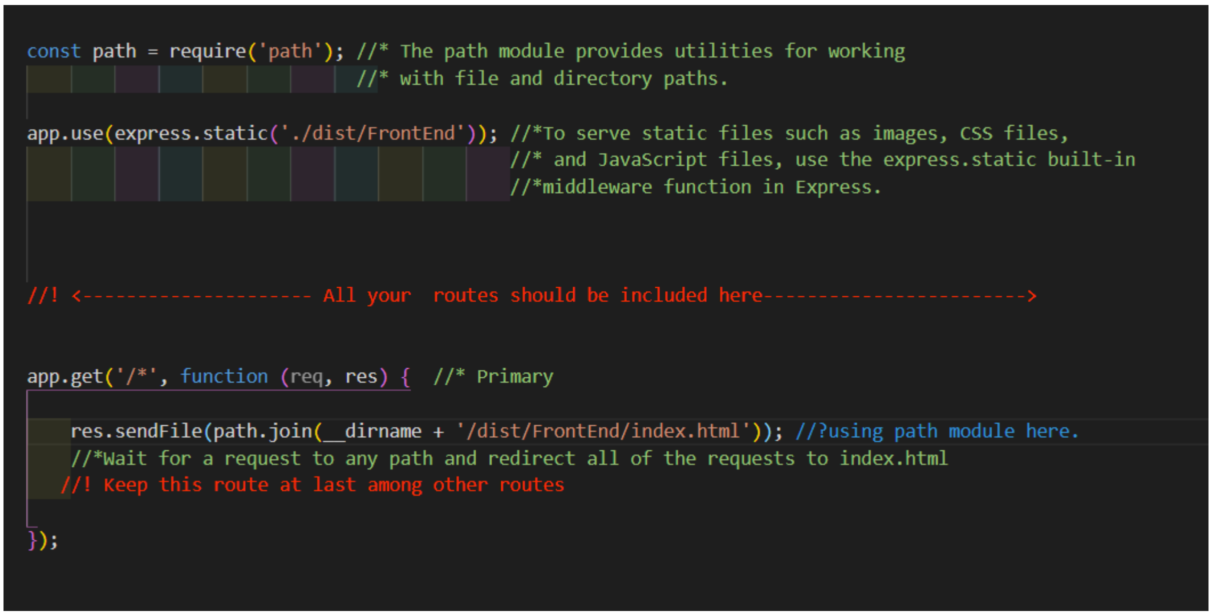
## II. Setting up your Back-end :

1. After copying `dist` folder from Frontend-folder to Backend-folder, in the app.js file add the following codes:

   - `const path = require('path');`
   - `app.use(express.static('./dist/`**`<folder-name inside dist>`**`'));`

   -
     ```
     app.get('/*', function(req, res) {
       res.sendFile(path.join(__dirname + '/dist//<folder-name inside dist>/index.html'));});
     });
     ```

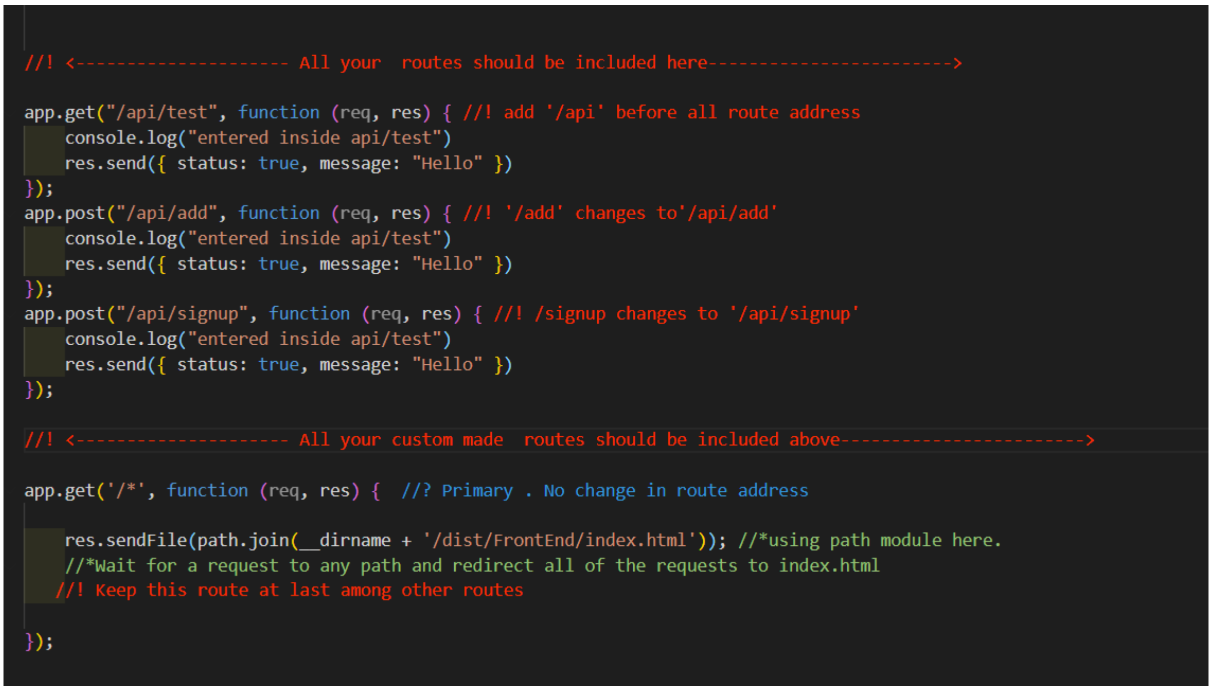
   `NB: in this case "folder-name inside dist" is FrontEnd`



```
const path = require('path'); //* The path module provides utilities for working
                              //* with file and directory paths.

app.use(express.static('./dist/FrontEnd')); //*To serve static files such as images, CSS files,
                                            //* and JavaScript files, use the express.static built-in
                                            //*middleware function in Express.


//! <-------------------- All your  routes should be included here---------------------->


app.get('/*', function (req, res) {  //* Primary

    res.sendFile(path.join(__dirname + '/dist/FrontEnd/index.html')); //?using path module here.
    //*Wait for a request to any path and redirect all of the requests to index.html
   //! Keep this route at last among other routes

});
```

2. Minor changes in the routes address

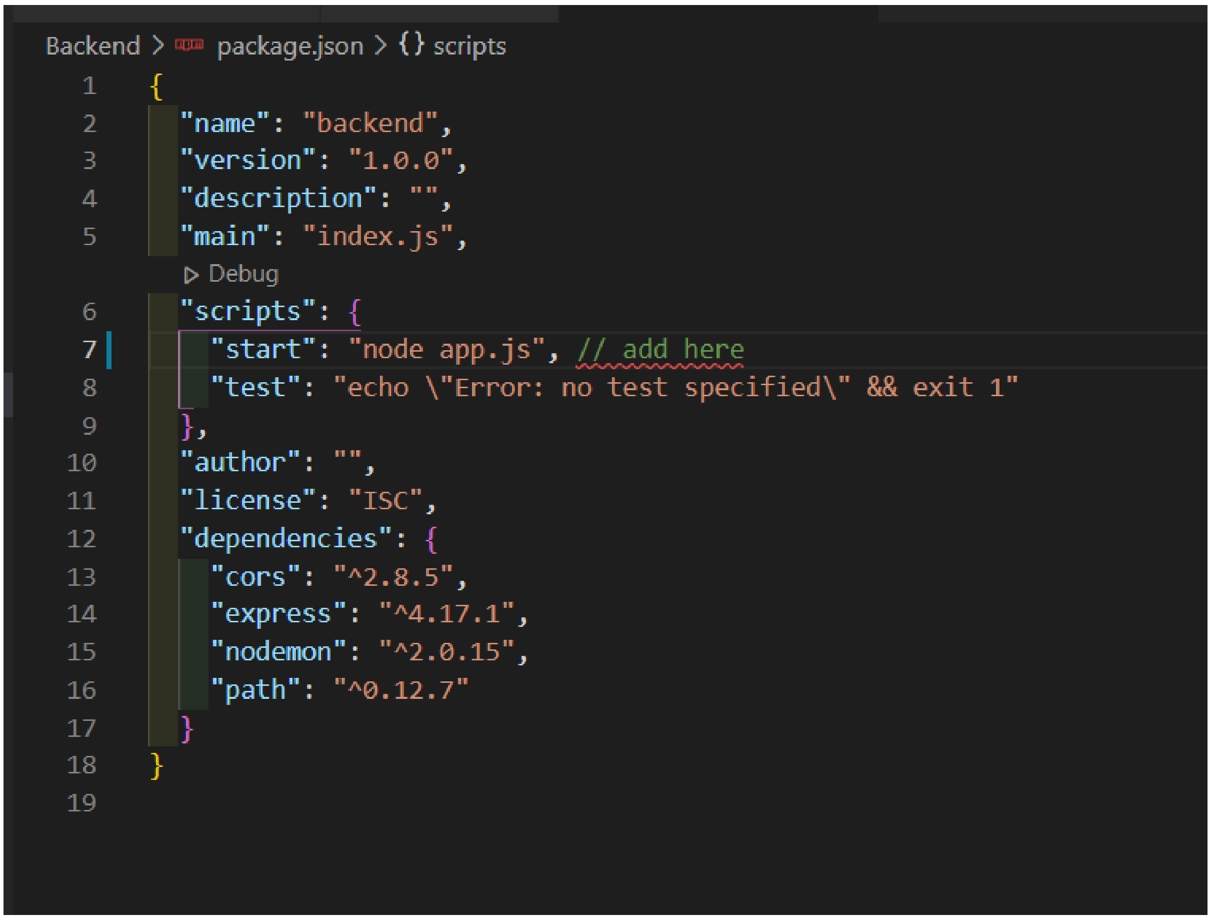
- Add `'/api'` before all router except Primary router as shown below



```
//! <--------------------- All your  routes should be included here----------------------->

app.get("/api/test", function (req, res) { //! add '/api' before all route address
    console.log("entered inside api/test")
    res.send({ status: true, message: "Hello" })
});
app.post("/api/add", function (req, res) { //! '/add' changes to'/api/add'
    console.log("entered inside api/test")
    res.send({ status: true, message: "Hello" })
});
app.post("/api/signup", function (req, res) { //! /signup changes to '/api/signup'
    console.log("entered inside api/test")
    res.send({ status: true, message: "Hello" })
});

//! <--------------------- All your custom made  routes should be included above----------------------->

app.get('/*', function (req, res) {  //? Primary . No change in route address

    res.sendFile(path.join(__dirname + '/dist/FrontEnd/index.html')); //*using path module here.
    //*Wait for a request to any path and redirect all of the requests to index.html
   //! Keep this route at last among other routes

});
```

3. In the package.json file of Backend-End Folder, add the following code inside the “scripts”:{}

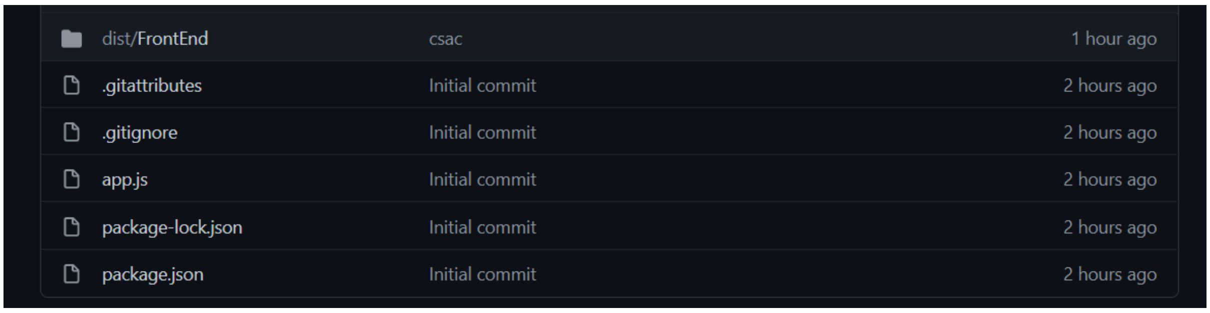
```
"start": "node app.js",
```



Backend > package.json > {} scripts

```
{
  "name": "backend",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  ▷ Debug
  "scripts": {
    "start": "node app.js", // add here
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "cors": "^2.8.5",
    "express": "^4.17.1",
    "nodemon": "^2.0.15",
    "path": "^0.12.7"
  }
}
```

4. Save it . Add the backend folder as a new repository in Github. (Make sure you have gitignored nodemodules)



|  |  |  |
|---|---|---|
| dist/FrontEnd | csac | 1 hour ago |
| .gitattributes | Initial commit | 2 hours ago |
| .gitignore | Initial commit | 2 hours ago |
| app.js | Initial commit | 2 hours ago |
| package-lock.json | Initial commit | 2 hours ago |
| package.json | Initial commit | 2 hours ago |

5. In the Heroku app, create a new app. Integrate it with your GitHub MEAN repository , you just created.

6. Deploy the project

7. If you find any error, use Heroku CLI to find where the error is .

   Reference : https://www.youtube.com/watch?v=U350rWtxGwg