Angular cli:-

Node –v

Npm –v

Npm install –g @angular/cli

Or

Npm I –g @angular/cli

Ng –v

After install ng

Create new project:

Ng new app1

Ngaap1> code .

It will open code in visual stidio code

App1>ng serve –open

The server will start server on port number 4200 and default browser will open automatically.

Ng test //run all the unit test

Ng e2e //runs all the end to end tests

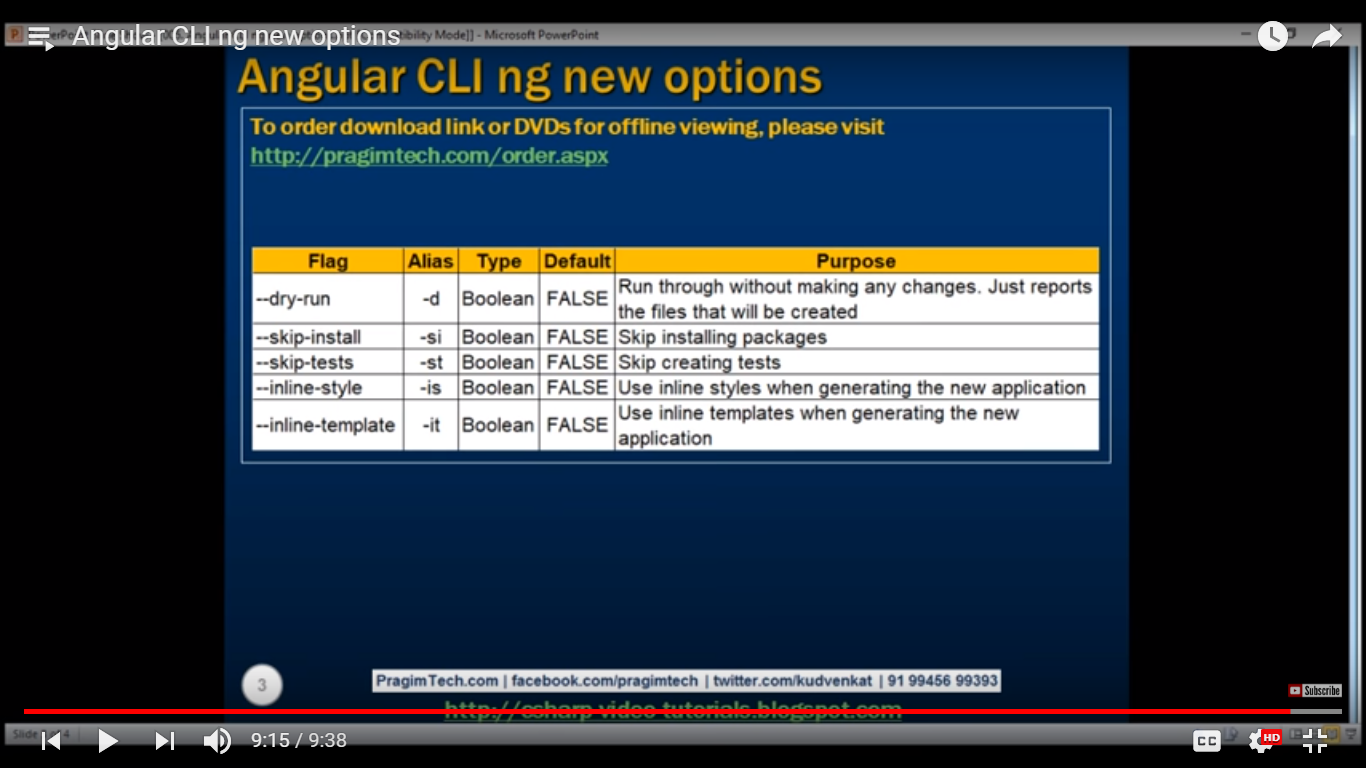
//kerma is angular 2 test runner and test is witten in language jasmine.

Ng –help

Ng –help | clip

And then open a notepad or word just past the prevous data

Ng –help > hlp.txt



Angular cli configuration file:-.angular-cli.json

{

"$schema": "./node\_modules/@angular/cli/lib/config/schema.json",

"project": {

"name": "app1" //Name of the project

},

"apps": [

{

"root": "src", //root folder of the project

"outDir": "dist", //output directory for build resource

"assets": [

"assets",

"favicon.ico"

],

"index": "index.html", //main html page of our application

"main": "main.ts", //main entry point file this file contains the code for bootstrap etc.

"polyfills": "polyfills.ts", //this is used for support every browser which is not support es6 or any other thing.

"test": "test.ts", //main entry point for unit test

"tsconfig": "tsconfig.app.json", //this file has data script compiler configuration

"testTsconfig": "tsconfig.spec.json", //data script compiler configuration for unit test.

"prefix": "app", //it is for all selector prefix. We can change this using ->ng g c xyz –-prefix myapp

"styles": [

"styles.css" //we can use css or scss using ->ng g c xyz -–style css/scss

],

"scripts": [],

"environmentSource": "environments/environment.ts",

"environments": {

"dev": "environments/environment.ts",

"prod": "environments/environment.prod.ts"

}

}

],

"e2e": {

"protractor": {

"config": "./protractor.conf.js"

}

},

"lint": [

{

"project": "src/tsconfig.app.json",

"exclude": "\*\*/node\_modules/\*\*"

},

{

"project": "src/tsconfig.spec.json",

"exclude": "\*\*/node\_modules/\*\*"

},

{

"project": "e2e/tsconfig.e2e.json",

"exclude": "\*\*/node\_modules/\*\*"

}

],

"test": {

"karma": {

"config": "./karma.conf.js"

}

},

"defaults": {

"styleExt": "css",

"component": {}

}

}

Package.json ->contains the packages to build and run our angular application.custom script can also be added.

{

"name": "app1",

"version": "0.0.0",

"license": "MIT",

"scripts": {

"ng": "ng",

"start": "ng serve", //when we use npm start it is used internally this. We can change this ex= ng serve –open it will also lunch the browser

"build": "ng build --prod",

"test": "ng test",

"lint": "ng lint",

"e2e": "ng e2e"

},

"private": true,

"dependencies": { //used to run the appliction

"@angular/animations": "^5.2.0",

"@angular/common": "^5.2.0",

"@angular/compiler": "^5.2.0",

"@angular/core": "^5.2.0",

"@angular/forms": "^5.2.0",

"@angular/http": "^5.2.0",

"@angular/platform-browser": "^5.2.0",

"@angular/platform-browser-dynamic": "^5.2.0",

"@angular/router": "^5.2.0",

"core-js": "^2.4.1",

"rxjs": "^5.5.6",

"zone.js": "^0.8.19"

},

"devDependencies": { //only for development purpose no need to these packages on our production server

"@angular/cli": "~1.7.2",

"@angular/compiler-cli": "^5.2.0",

"@angular/language-service": "^5.2.0",

"@types/jasmine": "~2.8.3",

"@types/jasminewd2": "~2.0.2",

"@types/node": "~6.0.60",

"codelyzer": "^4.0.1",

"jasmine-core": "~2.8.0",

"jasmine-spec-reporter": "~4.2.1",

"karma": "~2.0.0",

"karma-chrome-launcher": "~2.2.0",

"karma-coverage-istanbul-reporter": "^1.2.1",

"karma-jasmine": "~1.1.0",

"karma-jasmine-html-reporter": "^0.2.2",

"protractor": "~5.1.2",

"ts-node": "~4.1.0",

"tslint": "~5.9.1",

"typescript": "~2.5.3"

}

}

Node\_modules folder :- packages specified in the package.json file are installed into this folder.

E2e folder:- contains the end to end test and there configuration file.

.angular-cli.json:- algular cli configuration file.

.editorconfig :- configuration file for visual stidio code.we can share this file to other developer to write same style of code.

.gitignore :- files and folder listed in this file are ignored, when a change set to checked in to source control.

Karma.conf.js:- configuration file for karma (unit test runner)

protractor.conf.js:-configuration file for protector (e2e test framework for automation)

readme.md :- readme files which contains the commonly used angular cli commands out of the box.it contains help for common used commond.

Tsconfig.json:- typescript compiler configuration file.

{

"compileOnSave": false, //if u want to compile on save then use true

"compilerOptions": {

"outDir": "./dist/out-tsc",

"sourceMap": true,

"declaration": false,

"moduleResolution": "node",

"emitDecoratorMetadata": true,

"experimentalDecorators": true,

"target": "es5",

"typeRoots": [

"node\_modules/@types"

],

"lib": [

"es2017",

"dom"

]

}

}

Tslint.json:- configuration file for linting.it provide the compile time error for any error.

Scr folder:- used for create component pipes directory etc, other than this folder every thing is used for upport our development.

Src/assets:- this folder contains asset of the application like images any thing else to be copied when you build our application.

Src/Environment:-contains development and production environment file.

Favicon:favoret icon of application. It will shows on title of the application.

Index.html:- main html page. That will show first.

Main.ts:-this is the main entry file for our application.this file contains the code of the application.

Polyfiles.ts:contains polyfiles needed by angular.

Styles.css:- contain the golabal styles for our application.

Test.ts:- this is the main entry point of the unit test. This file load all spec files for unit test.

Typing.d.ts:-type script typing file. Typescript editor leverage these type definition file to display type information.it give error at editor if any error.

App.module.ts:- root application module.

**Create component:-**when we create the component automatically it generates 4 files and one folder of that component name and component is also registered in our module.ex:

E:\rahul\angular\app1>ng g c app1

create src/app/app1/app1.component.html (23 bytes)

create src/app/app1/app1.component.spec.ts (614 bytes)

create src/app/app1/app1.component.ts (261 bytes)

create src/app/app1/app1.component.css (0 bytes)

update src/app/app.module.ts (390 bytes)

for create withis other folder:

E:\rahul\angular\app1>ng g c app1\xyz

Or

E:\rahul\angular\app1>ng g c app1 – flat // directolly place the all folder in app folder

To generate service:

>ng g s servicename

E:\rahul\angular\app1>ng g s customer

create src/app/customer.service.spec.ts (386 bytes)

create src/app/customer.service.ts (114 bytes)

E:\rahul\angular\app1>ng g s customer1 -m=app.module //also update module.ts

create src/app/customer1.service.spec.ts (392 bytes)

create src/app/customer1.service.ts (115 bytes)

update src/app/app.module.ts (462 bytes)

E:\rahul\angular\app1>ng g s customer1 --spec=false -d

create src/app/customer1.service.ts (115 bytes)

ng generate module modulename or

ng g m modulename

ng generate directive diretivename or

ng g d directivename

ng generate class classname

ng g cl classname

ng generate interface interfacename

ng g I interfacename

ng generate enum enumname

ng g e enumname

Linting:

