Angular-8

Angular CLI Part-1

Presentation BY RAJ PRUDHVI

Contact us:https://training.uplatz.com
Email: info@uplatz.com
Phone: +44 7836 212635

Uplatz

Angular CLI

- Introduction to Angular CLI
- Installing Angular CLI
- Angular CLI create new project
- Angular CLI ng new options
- Angular CLI configuration file
- Angular CLI project structure

Uplatz

Angular CLI

- If you have any experience with Angular, then you already know manually setting up an Angular application from scratch is a laborious and time consuming process. We have to

  - ✓ Create the root application module (AppModule) as every angular application should have atleast one module which is the root module
  - ✓ Create main.ts file which is the entry point to the application. The code in this file, loads the angular root module – AppModule
  - ✓ Create index.html which hosts our application

You have to manually write all the boilerplate code yourself, which is not only monotonous but also time consuming.

Angular CLI

- In a real world, we usually have more than one developer working on a given angular project. While all these developers are creating these different files and writing the required boiler plate code, are they following the angular teams best practices and conventions. What if the developers are not following them. How do we enforce them?

- The option is to have some tooling in place to address this. Angular CLI is such a tool. It help us create angular applications, components, modules, pipes, directives, services and much more with great speed and consistency while still following the angular teams best practices and conventions.

What is Angular CLI

CLI stands for Command Line Interface.
So Angular CLI is command line tool they help us

- Create Angular applications faster and with great consistency
- Create the boiler plate code for angular features like components, directives, pipes, services etc.
- Create boiler plate code for Typescript features like classes, interfaces, enums etc.
- It follows angular best practices and conventions out of the box
- Run Unit and End-to-End (e2e) tests
- Create optimized builds for production

Installing Angular CLI

- The prerequisites for installing Angular CLI : To install Angular CLI you should have installed Node 6.9.0 or higher, and NPM 3 or higher
- To check the versions that you have on your machine type the following commands in a command window.

  - node -v
  - npm –v

- You can get the latest version of Node and NPM from the following website. Click on the correct download link depending on the Operating System you have.

  - https://nodejs.org/en/download/

Installing Angular CLI

- Once you have Node and NPM installed. Run Command Prompt and execute the following command. Flag -g installs Angular CLI globally on your machine.

```
npm install -g @angular/cli
```

- You can also use i as shortcut for install. So the above command can also be rewritten as shown below

```
npm i -g @angular/cli
```

Installing Angular CLI

- To verify the version of Angular CLI installed, execute the following command

```
ng v
```

- If you run into any problems installing Angular CLI, follow these steps and hopefully Angular CLI will be installed successfully.

Step 1 : Delete "npm" folder from the following path

C:\Users\Your_UserName\AppData\Roaming

Please note : If you cannot find "AppData" folder, make sure in your windows operating system, you have "Show hidden files, folders, and drives" option is turned on. "AppData" is a hidden folder.

Uplatz

Installing Angular CLI

Step 2 : Once you have the "npm" folder deleted, uninstall node.js. On a windows machine you can uninstall node.js from Control Panel\All Control Panel Items\Programs and Features. Right click on "Node.js" and select "uninstall" from the context menu.

Step 3 : Reinstall Node.js by downloading the appropriate installer for your operating system from the following link. https://nodejs.org/en/download/

Step 4 : Run Command Prompt and try to install Angular CLI again using the following command.

Angular CLI Create New Project

To create a new Angular Project, open Command Prompt and execute the following command. This command creates all the required files and also installs all the required packages. Depending on your computer and internet connection speed, this command takes a few minutes to complete.

ng new AngularApp1

- ng is the Angular CLI
- new for creating a new application
- AngularApp1 is the name of your angular application

Uplatz

Angular CLI Create New Project

So what did this "ng new" command do

- A new folder with name AngularApp1 is created
- All the required configuration and source files are created.
- All the npm dependencies are installed in node_modules folder
- Unit and end-to-end tests are created
- The Karma unit test runner is configured
- The Protractor end-to-end test framework is configured

Run Angular Project

To run the project using Angular CLI, type the following command at the command prompt. This command builds the application and opens it in our default browser.

```
ng serve --open
```

The flag --open, launches our default browser and runs the application. By default the application runs on port 4200. We can change this port number if required.

At the moment, the angular development server is running in watch mode, meaning when a file changes, those changes are automatically detected, compiled and the browser reloads to reflect the changes. This is called live reload. We can turn this live reload functionality off, if required.

Stop Server , Run Unit Tests , End-End Tests

- To stop the server, press CTRL + C while you are on the command prompt and then "Y" and ENTER key. This will stop the server.

- To run all the unit tests, use the following command

  ng test

- To run all the end-to-end tests, use the following command

  ng e2e

Angular CLI ng new options

Flag	Type	Default	Alias	Purpose
--dry-run	Boolean	false	-d	Run through without making any changes. Just reports the files that will be created
--skip-install	Boolean	false		Skip installing packages
--skip-tests	Boolean	false	-st	Skip creating tests
--inline-style	Boolean	false	-is	Use inline styles when generating the new application
--inline-template	Boolean	false	-it	Use inline templates when generating the new project

Angular CLI configuration file

angular.json : This is the configuration file that the Angular CLI uses. it has several settings in it

The settings from this file are used when we

- Generate angular features likes components, pipes, services etc
- Run unit and end-to-end tests
- Build the application etc.

Angular CLI configuration file

The important point to take away is that the values in the Angular CLI configuration file depends on the options that you have used with the "ng new" command when generating a new angular project.

For example, if you do not use the --prefix option with the "ng new" command, then the default value "app" is stored in the configuration file for "prefix" setting. So the root component (AppComponent) that is created at the application generation time has "app" as the selector prefix.

Angular CLI configuration file

Instead if you want “uplatz" as the prefix, use --prefix flag along with "ng new“ command. When you do this several things happen

- “uplatz" is stored as the "prefix" setting value in .angular.json configuration file
- "uplatz" is used as the selector prefix for the root component that the "ng new“ command generates
- Any new component that you generate in the future using the following command will also have “uplatz" as the selector prefix

ng generate component componentName

Uplatz

Angular CLI configuration file

- If you want to override the prefix setting in the angular cli configuration file, you can use **--prefix** option with the generate command as shown below.

ng generate component abc --prefix raj

- This will generate the component “abc" with the prefix “raj" instead of “uplatz“

Some of the options like --prefix can be used with several commands like ng new and ng generate

Uplatz

Angular CLI Project Structure

package.json : This file contains the packages to build and run our application. It contains two sets of packages, dependencies and devDependencies. The dependencies are essential for running the application. The devDependencies are only required to develop the application. These packages are installed into the node_modules folder by the Node Package Manager (npm), when npm install command is executed. You can also add your own custom scripts here.

"scripts" property in package.json file contains the useful npm commands.
Notice we have **"start": "ng serve".**
This means when we execute **npm start** it executes **ng serve** which builds and starts the server.

In addition if you also want to launch the browser and open the application
change "start": "ng serve“ to **"start": "ng serve --open"**

Angular CLI Project Structure

node_modules : The packages specified in package.json file are installed into this folder when we run npm install command

e2e folder : Contains end-to-end tests and their configuration files

angular.json : This is the Angular CLI configuration file

.editorconfig : Configuration file for Visual Studio Code. The settings in this file let you set certain code style guidelines.

.gitignore : This file is used to determine files and folders you don't want to check in to source control. For example one of the folders we do not want to check in to source control is /dist folder which is auto generated when we build the application. So this folder is listed in this file. So, all the files and folders listed in this file are ignored, when a change set is checked in to source control.

Angular CLI Project Structure

karma.conf.js : Karma is the unit test runner for angular applications. As the name implies, karma.conf.js is the configuration file for Karma.

protractor.conf.js : Protractor is an end-to-end test framework for Angular applications. As the name implies, protractor.conf.js is the configuration file for Protractor.

README.md : This is a README file which contains the commonly used Angular CLI commands out of the box. You may enhance it with your own project documentation so that anyone checking out the repo knows the commands to use to build, run and test your app.

tsconfig.json : This is the Typescript compiler configuration file. This file has several Typescript compiler configuration settings.
For example, to compile Typescript to JavaScript on saving a Typescript file set compileOnSave setting to true.

Angular CLI Project Structure

tslint.json : Angular has a linting tool that checks our Typescript code for programmatic and stylistic errors as well as non-adherence to coding standards and conventions. tslint.json is the configuration file for linting.

src folder : As the name implies, this folder contains all our angular project source code. Components, templates, pipes, services, images, styles etc that our angular application needs are present in this folder. The rest of the files and folders that are present outside this folder, are there to support building our angular application

assets : As the name implies, the assets folder contains the assets of your application like images and anything else to be copied when you build your application

Angular CLI Project Structure

environments: This folder contains the environment files. By default we have 2 environment files. **environment.ts** is for for development environment. Notice production property in this file is set to false. **environment.prod.ts** is for production. Notice in this file production property is set to true as expected. The build system defaults to the dev environment which uses `environment.ts`, but if we do a production build environment.prod.ts will be used. The file and environment mapping is in Angular CLI configuration file (angular.json)

favicon.ico : This is the favorite icon for your application which is typically displayed in the browser address bar and next to the page name in a list of bookmarks. Angular CLI provides this favorite icon out of the box. You may replace this favicon with your own company favicon

index.html : The main HTML page that is served when someone visits your site

Angular CLI Project Structure

main.ts : The main entry point for the application. This file contains the code to bootstrap the application root module (AppModule)

polyfills.ts : This is the polyfills file. Angular is built on the latest standards of the web platform. Targeting such a wide range of browsers is challenging because not all browsers support all features of modern browsers. This can be compensated by using polyfill scripts as they implement the missing features in JavaScript. So these polyfills allow us to use an API regardless of whether it is supported by a browser or not

styles.css : This file contains the global styles of our application. Styles that are local and specific to a component are often defined with in the component itself for easier maintenance

test.ts : This file is the main entry point for unit tests and loads all the .spec and framework files

Angular CLI Project Structure

tsconfig.app.json : Typescript compiler configuration for the Angular app

tsconfig.spec.json : Typescript compiler configuration for the unit tests

app folder : The root component (AppComponent) Typescript, HTML template, Style Sheet and Spec files are present inside app folder

app.module.ts : This is the root application module (AppModule)

Thank You

Contact us:https://training.uplatz.com

Email: info@uplatz.com

Phone: +44 7836 212635

Uplatz