Angular 2 Tutorial

1. Angular JS is an open source framework built over JavaScript.
2. It was built by the developers at Google.
3. This framework was used for Single Page applications.

## **Features of Angular 2**

Following are the key features of Angular 2 –

* **Modules** – A module is a logical piece of code which is designed to perform a single task.
* **Component** − In Angular 1 we had controller while from Angular 2 onwards we have components. A single component can multiple module together.
* **Templates** − This is used to define the views of an Angular JS application.
* **Metadata** − This can be used to add more data to an Angular JS class.
* **Service** − This is used to create components which can be shared across the entire application. For example, if you had a data component that picked data from a database, you could have it as a shared service that could be used across multiple applications.
* **TypeScript** − The newer version of Angular is based on TypeScript. This is a superset of JavaScript and is maintained by Microsoft.

# Angular 2 - Environment

To start working with Angular 2, you need to get the following key components installed.

* **Node.js and Npm –** To get started with Angular js, we will need to have Node.js andNPM known as the node package manager is installed as a part of Node.js.

Angular JS as a framework has dependencies on other components. And Npm can be used to download these dependencies and attach them to your project.

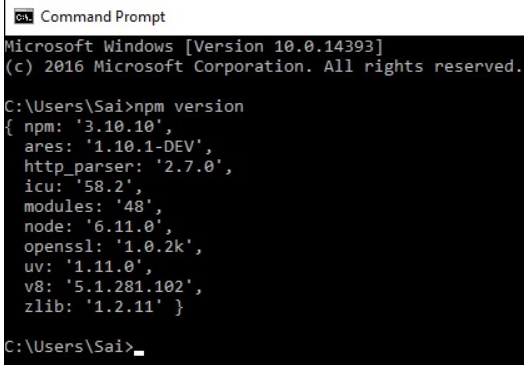
**$ npm -v**: To check if Npm is available, we can open a terminal window and type **$ npm -v**

* **TypeScript:** Once we have Node.js setup the next step is to install TypeScript. To install it run the following npm command

**$ npm install -g typescript**

**Note:** We don’t have to use TypeScript to use Angular, but we probably should. Angular does have an ES5 API but Angular is written in TypeScript.

* **Git −** This is the source code software that can be used to get the sample application from the GitHub angular site.
* **Editor −** There are many editors that can be used for Angular JS development such as Visual Studio code and WebStorm. In our tutorial, we will use Visual Studio code which comes free of cost from Microsoft.
* To confirm the installation, in the command prompt you can issue the command npm version. You will get the version number of npm as shown in the following screenshot.



* **Angular CLI:** It provides a utility to create and manage projects from the command line. It automates tasks like creating projects, adding new controllers etc. It’s generally good idea to use Angular CLI as it helps in creating and maintaining common patterns across our application.

**To install Angular CLI, just run the following command.**

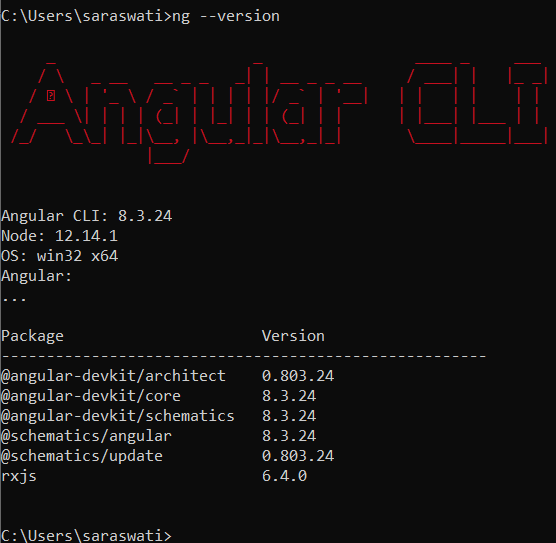
$ npm install -g @angular/cli

Once it’s installed we will be able to run it from the command line using **ng** command

**To see the current version of Angular CLI:**

**$ ng –version:** and we will get following output a shown in the screen.

|  |
| --- |
| **C:\Users\saraswati>ng --version**  \_ \_ \_\_\_\_ \_ \_\_\_  / \ \_ \_\_ \_\_ \_ \_ \_| | \_\_ \_ \_ \_\_ / \_\_\_| | |\_ \_|  / △ \ | '\_ \ / \_` | | | | |/ \_` | '\_\_| | | | | | |  / \_\_\_ \| | | | (\_| | |\_| | | (\_| | | | |\_\_\_| |\_\_\_ | |  /\_/ \\_\\_| |\_|\\_\_, |\\_\_,\_|\_|\\_\_,\_|\_| \\_\_\_\_|\_\_\_\_\_|\_\_\_|  |\_\_\_/  Angular CLI: 8.3.24  Node: 12.14.1  OS: win32 x64  Angular:  ...  Package Version  **------------------------------------------------------**  @angular-devkit/architect 0.803.24  @angular-devkit/core 8.3.24  @angular-devkit/schematics 8.3.24  @schematics/angular 8.3.24  @schematics/update 0.803.24  rxjs 6.4.0  **C:\Users\saraswati>** |

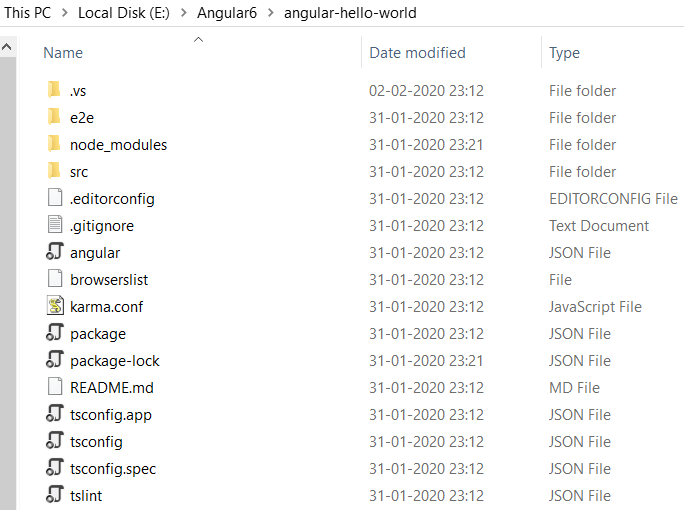


**$ ng –help:** Provide all Angular CLI command.

**Creating Angular Project using CLI:** Following command is used to create angular project.

**$ ng new angular-hello-world:** it will create an Angular Project with name **‘angular-hello-world’**

**Created Project:**



|  |  |
| --- | --- |
| e2e | end-to-end tests gets created during project creation through CLI. This is nothing but integration testing which make sure that our application function correctly |
| node\_modules | The **node\_modules folder** contains libraries downloaded from npm. You shouldn't push it to GitHub (you should even add it to your. gitignore), because everyone who clones your repository can download it themselves (based on your package.json).  We **could remove** your **node\_modules**/ **folder** and then reinstall the dependencies from package.json. This would erase all installed packages in the current **folder** and only install the dependencies mentioned in **package. json.**  The **node\_modules** directory is only for build tools. The package.json file in the app root defines what libraries will be installed into **node\_modules** when you run npm install |
| Src | This folder contains the main code files related to our angular application |
| angular.json | Angular cli configuration file |
| browserlist | Provide the list of browsers supported |
| package.json | Pacakage.json file gets created during the project creation and it maintains all the bunch of dependencies required for the project. Initially during project creation, it includes two dependencies required for running(**dependencies)** and development(**devDependencies)** of the project and but later as soon as we include other functionalities in the project all the dependencies gets added in the package.json file. |
| package-lock.json | The This is mainly used to **lock** dependencies to a specific version number defined in package.json. |
| README.md | A useful README |
| tsconfig.app.json |  |
| tsconfig.json | Type Script config |
| tsconfig.spec.json |  |
| Tslint.json | Linting config |

**Package.json**

|  |  |
| --- | --- |
| {  "name": "angular-hello-world",  "version": "0.0.0",  "scripts": {  "ng": "ng",  "start": "ng serve",  "build": "ng build",  "test": "ng test",  "lint": "ng lint",  "e2e": "ng e2e"  },  "private": true,  "**dependencies**": {  "@angular/animations": "~8.2.14",  "@angular/common": "~8.2.14",  "@angular/compiler": "~8.2.14",  "@angular/core": "~8.2.14",  "@angular/forms": "~8.2.14",  "@angular/platform-browser": "~8.2.14",  "@angular/platform-browser-dynamic": "~8.2.14",  "@angular/router": "~8.2.14",  "rxjs": "~6.4.0",  "tslib": "^1.10.0",  "zone.js": "~0.9.1"  }, | "**devDependencies**": {  "@angular-devkit/build-angular": "~0.803.24",  "@angular/cli": "~8.3.24",  "@angular/compiler-cli": "~8.2.14",  "@angular/language-service": "~8.2.14",  "@types/node": "~8.9.4",  "@types/jasmine": "~3.3.8",  "@types/jasminewd2": "~2.0.3",  "codelyzer": "^5.0.0",  "jasmine-core": "~3.4.0",  "jasmine-spec-reporter": "~4.2.1",  "karma": "~4.1.0",  "karma-chrome-launcher": "~2.2.0",  "karma-coverage-istanbul-reporter": "~2.0.1",  "karma-jasmine": "~2.0.1",  "karma-jasmine-html-reporter": "^1.4.0",  "protractor": "~5.4.0",  "ts-node": "~7.0.0",  "tslint": "~5.15.0",  "typescript": "~3.5.3"  }  } |

* lists the packages your project depends on
* specifies versions of a package that your project can use
* Three types of dependencies we can add in **package. json**

1. Dependency
2. DevDependency
3. peerDependency
4. **dependencies**: Basically, required for running angular application

**Note1"@angular/......:** is mandatory packages required for angular application development

**Note2**- **rxjs": "~6.4.0**: is a third-party library which helps frontend to connect with backend services (e.g. rest API calls) through http connections. So, these are the support packages which are not mandatory for development angular applications.

1. **devDependencies:** Basically, required for developing angular application

This include angular and third-party libraries which is important for developing an application. E.g. **@angular/compiler-cli** is required for interpreting type script into java script and help in running on the browsers.

We don’t need **devDependencies** in production but **dependencies** are required in the production env**.**

**Versioning in package. Json**

**Note3**- : Here we can see that in the version part some version starts with (~ tild) and few starts with (^ carrot) symbol. And this version is in the form of x.y.z e.g. (**6.4.0)** . X is the major version Y is the Minor version and Z is the patch version.

**(~ tilde):** it says installed any thing which is equal to 6.4.0 or the package with the higher patch version(z) e.g. (**6.4.1**). it means variation of version will be in z part.

**(^ carrot):** Install anything with the same in major(X) but with higher in minor (Y) or latest in patch(Z). **e.g. (6.5.1)**

**package-lock.json:**

The **package**.**json** is used **for** more than dependencies - like defining project properties, description, author & license information, scripts, etc.

The **package**-**lock**.**json** is solely used to **lock** dependencies to a specific version number.

package.json is a file that contains information about your project (name, version, etc) and it lists the packages that your project is dependent on.

package-lock.json will simply avoid this general behavior of installing updated minor version so when someone clones your repo and run npm install in their machine. NPM will look into package-lock.json and install exact versions of the package as the owner has installed so it will ignore the ^ and ~ from package.json.

**angular.json**

* 1. Before angular 6 this file come with [angular-cli.json]. In new angular version [angular-cli.json] has been replaced by angular.json
  2. This file generally configures all the src related file like (favicon, index.html, main.ts, styles.css).
  3. We can configure several project src folder file under one angular.json while earlier in angular-cli.json it was not possible.

|  |  |
| --- | --- |
| {  "$schema": "./node\_modules/@angular/cli/lib/config/schema.json",  "version": 1,  "newProjectRoot": "projects",  "projects": {  "angular-hello-world": {  "projectType": "application",  "schematics": {},  "root": "",  "sourceRoot": "src",  "prefix": "app",  "architect": {  "build": {  "builder": "@angular-devkit/build-**angular:browser**",  "options": {  "outputPath": "dist/angular-hello-world",  "index": "src/index.html",  "main": "src/main.ts",  "polyfills": "src/polyfills.ts",  "tsConfig": "tsconfig.app.json",  "aot": false,  "assets": [  "src/favicon.ico",  "src/assets"  ],  "styles": [  "src/styles.css"  ],  "scripts": []  },  "configurations": {  "production": {  "fileReplacements": [  {  "replace": "src/environments/environment.ts",  "with": "src/environments/environment.prod.ts"  }  ],  "optimization": true,  "outputHashing": "all",  "sourceMap": false,  "extractCss": true,  "namedChunks": false,  "aot": true,  "extractLicenses": true,  "vendorChunk": false,  "buildOptimizer": true,  "budgets": [  {  "type": "initial",  "maximumWarning": "2mb",  "maximumError": "5mb"  },  {  "type": "anyComponentStyle",  "maximumWarning": "6kb",  "maximumError": "10kb"  }  ]  }  }  }, | ------------------------------------------------------------  "serve": {  "builder": "@angular-devkit/build-angular:dev-server",  "options": {  "browserTarget": "angular-hello-world:build"  },  "configurations": {  "production": {  "browserTarget": "angular-hello-world:build:production"  }  }  },  "extract-i18n": {  "builder": "@angular-devkit/build-angular:extract-i18n",  "options": {  "browserTarget": "angular-hello-world:build"  }  },  ------------------------------------------------------------  "test": {  "builder": "@angular-devkit/build-angular:karma",  "options": {  "main": "src/test.ts",  "polyfills": "src/polyfills.ts",  "tsConfig": "tsconfig.spec.json",  "karmaConfig": "karma.conf.js",  "assets": [  "src/favicon.ico",  "src/assets"  ],  "styles": [  "src/styles.css"  ],  "scripts": []  }  },  "lint": {  "builder": "@angular-devkit/build-angular:tslint",  "options": {  "tsConfig": [  "tsconfig.app.json",  "tsconfig.spec.json",  "e2e/tsconfig.json"  ],  "exclude": [  "\*\*/node\_modules/\*\*"  ]  }  },  -----------------------------------------------------------  "e2e": {  "builder": "@angular-devkit/build-angular:protractor",  "options": {  "protractorConfig": "e2e/protractor.conf.js",  "devServerTarget": "angular-hello-world:serve"  },  "configurations": {  "production": {  "devServerTarget": "angular-hello-world:serve:production"  }  }  }  }  }},  "defaultProject": "angular-hello-world"  } |

For more reference: <https://www.youtube.com/watch?v=rT_j1UgjM4k>