Setup Angular Workspace

- Angular workspace comprises of a set of Angular Projects. [Angular Applications]
- You can configure a workspace for all projects.
- Workspace will maintain a common repository [library] for all your project.
- You can access and use the repository across all projects, instead of installing and downloading for every project individually.
- To configure workspace run the following command
  C:\> ng new Your-Workspace-Name --createApplication=false
  C:\> ng new Angular7pm-Workspace --createApplication=false
- This will create a new folder on your physical drive **"Angular7pm-Workspace"**
- Open the workspace folder in Visual Studio Code
- Angular workspace comprises of following components initially

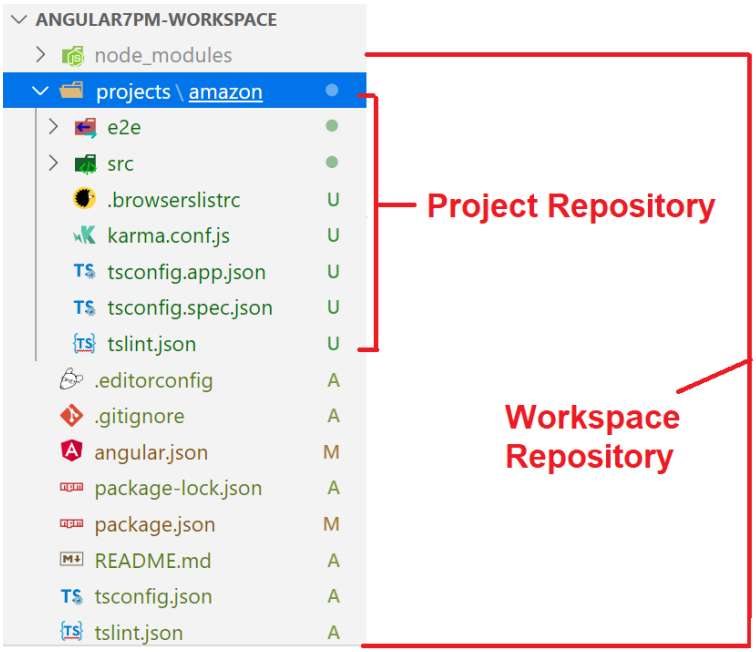
Component	Description
node_modules	- It is a folder that comprises of library files installed by using NPM [Package Manager]. - Library can be like bootstrap, jQuery, animations, materials, angular etc. - This library will be shared to all projects in the current workspace.

.editorconfig	- It is a configuration file for code editors. - Multiple developers working on same project across various editors and IDE's may have different configuration, which leads to code consistency issues. - EditorConfig file allows to configure common rules for coding. - Various editors used for angular are **Visual Studio Code** **PyCharm** **Webstrom** **Sublime** **RubyMine,** **Visual Studio,** **Eclipse etc..**
.gitignore	- Specifies intentionally untracked files that Git should ignore.
angular.json	- CLI configuration for all projects in the workspace. - Configuration include options for building, serving, testing etc.
package.json	- Package is a collection of libraries. - Few libraries are not individual they have dependencies. - This will configure NPM package dependencies that are available to all project in workspace. - By using this package.json you can extract and install all dependences.
package-lock.json	- Provides the version information for all packages that are installed into "node_modules" - Version information is required to

	verify the package available and package installed and update packages.
README.md	- Help document
tsconfig.json	- TypeScript configuration file - Allows to configure the target JS version like ES5, ES6. - It also configures the output direction.
tsLint.json	- It is language analysis tool. - It sets the rules for TypeScript. - Data Type, code block etc.

Create a new Angular Application

- Open Terminal with your workspace folder
- Type the following command
  C:\Angular7pm-Workspace>ng generate application amazon
  Would you like to share anonymous data information with Angular Team? No
  Would you like to add Angular routing? No
  Which stylesheet format would you like to use? CSS [Less, SCSS etc.]
- This will create a new folder "**projects**" in workspace and add "**amazon"** project folder.
- Every project comprises for several components



Note: To Start project of server

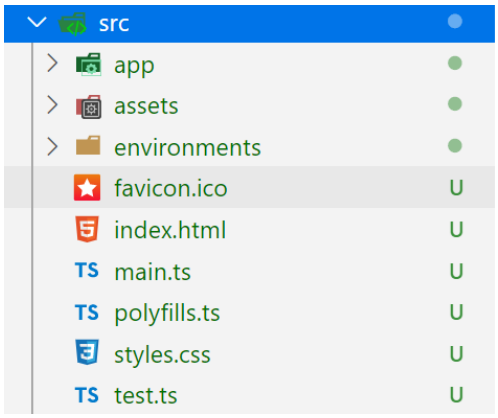
> ng serve --project=amazon
You application starts of http://localhost:4200

- **Angular application comprises of the following files and folders at hight level**

File/Folder	Description
e2e	- It is End-to-End configuration folder. - It contains sources files for a set of end-to-end-tests.
src	- Source files for the root level application. - It comprises both dynamic and non-dynamic files. - Application resources like component, services, pipes, modules etc. - CSS, HTML etc.
.browserslistrc	- The config to share target browser and Node.js versions for different front-end tools. [plugins]
karma.conf.js	- Application specific karma configuration

	file. - It is a testing framework used in Angular.
tsconfig.app.json	- Application specific TypeScript configuration file.
tsconfig.spec.json	- “spec” is testing specification. - TypeScript configuration for application tests.
tslint.json	- Application specific TSlint.

Application related all source file are present in “src” folder



Application Source	Description
app/	- It contains the component files in which your application logic and data are defined. - You can define components, directives, services, modules, pipes etc.
assets/	- It comprises of non-dynamic files like, images, text, pdf and other static resources.
environments/	- Entire development process comprises of build information for development, Quality and production server. - Contains build configuration options for a particular target environment like development, production etc.
favicon.icon	- An icon used for bookmarking your

	application in browser. <head> <link rel=”shortcut icon” href=”favicon.icon”> </head>
index.html	- Application starts with “index.html” - It is the start up page. - It is the first page to be served. - In SPA user will access only index.html and he can browse through entire application from index page.
main.ts	- The main entry point for your application. - Compiles the application with the JIT compiler and bootstraps the application. - It creates a chunk that compiles and converts the static DOM into dynamic DOM. - Angular 9 and 10 by default uses “AOT” complier. - You can define --aot while using “server” command.
pollyfills.ts	- Provides the scripts for browser support. - It uses differential loading technique. - It loads legacy and modern scripts according to browser. - Legacy Browser: HTML 4 and ES4 - Modern Browser: HTML 5 and ES5
styles.css	- It comprises of global styles. - It can import and configure styles, which are accessible across all components in application.
test.ts	- It is the main entry point for testing.

	- Your application starts Unit Testing from this location. - It responsible for creating a Test Bed and configure the test to run.

Setup Bootstrap CSS for Angular Application

- **Download and Install Bootstrap for your Workspace**
  C:\Angular7pm-Workspace>npm install bootstrap

```
node_modules
|
|_bootstrap
  |
  |_dist
    |_css
      |_bootstrap.css / bootstrap.min.css
```

- Go to “styles.css” file in “src” and import bootstrap.css
  @import '../../../node_modules/bootstrap/dist/css/bootstrap.css';

Download and Use Fontawesome

- Download “Fontawesome” Free for Web.
- Extract and copy all files.
- Go to “node_modules” folder in file explore.
- Create a sub folder by name “Fonts”
- Paste all fontawesome files into the sub folder.
- Import in “styles.css”
  @import '../../../node_modules/Fonts/css/all.css';

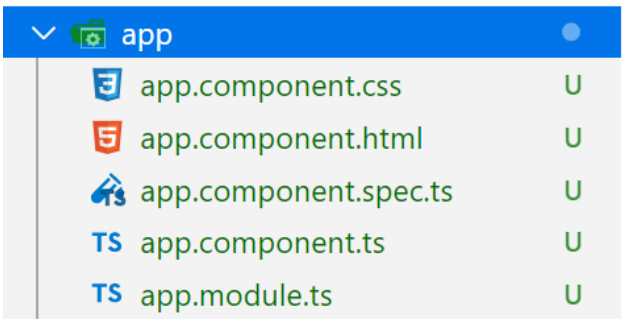
Angular Application Folder

- Angular application comprises of
  - Components
  - Services

- Pipes
- Directives
- Modules
- Factory etc.

Component	It defines a template that comprises of presentation and functionality.
Service	It defines business logic, which you can use across components.
Pipe	It is used to transform data, It handles the functions like formatting, filtering, sorting etc.
Directive	It is responsible for extending HTML, converts the static DOM element into Dynamic.
Factory	It is a collection of related type of functions.
Module	It is a combination of component, services, directives, pipes etc.

- The “app” folder comprises of all component, directives, pipes, services and modules that are used for application.



- Every Angular application by default provided with a sample component called “AppComponent”
- Angular Application uses “AppComponent” as start-up component.
- Index.html page starts with “AppComponent”.
- Every component in Angular comprises of 4 files

    - app.component.ts : Logic
    - app.component.html : Presentation
    - app.component.css : Styles
    - app.component.spec.ts : Unit Tests
- Every angular application comes with a module "**app.module.ts"**

app.module.ts

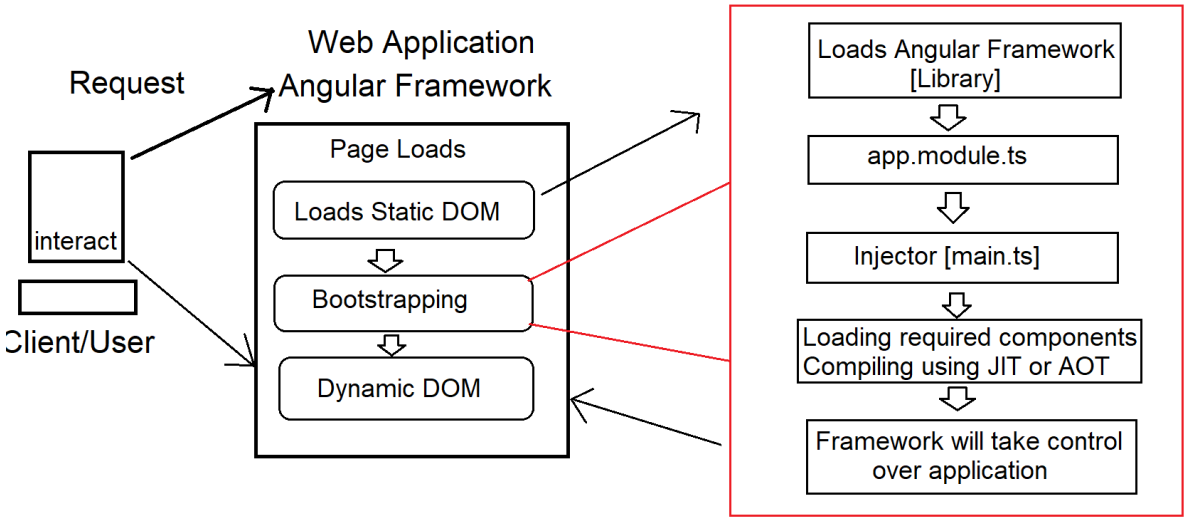
- Your application comprises of components, services, pipes, modules etc.
- If application have to use the components and other contents then it must import and load on application start.
- Which component or content to import and load is decided in "**app.module.ts**".
- If any content is not registered with "app.module.ts" and imported then you can't use in application.
- It uses "**Dependency Injection"** which identifies the dependencies required for that particular situation and loads them into memory.
- It comprises of your application "meta data".
- "main.ts" is compiling your application, it verifies the dependencies from **"app.module.ts"** and imports them, then load into memory.
- Angular module is class that comprises various settings required for application.
- It implements the members from "NgModule".
- Module related all members are derived for "NgModule" defined in **"@angular/core".**
- **@NgModule()** is decorator for module.
- **@Directive()** is decorator for directive.
- **@Component()** is decorator for component.

Syntax:

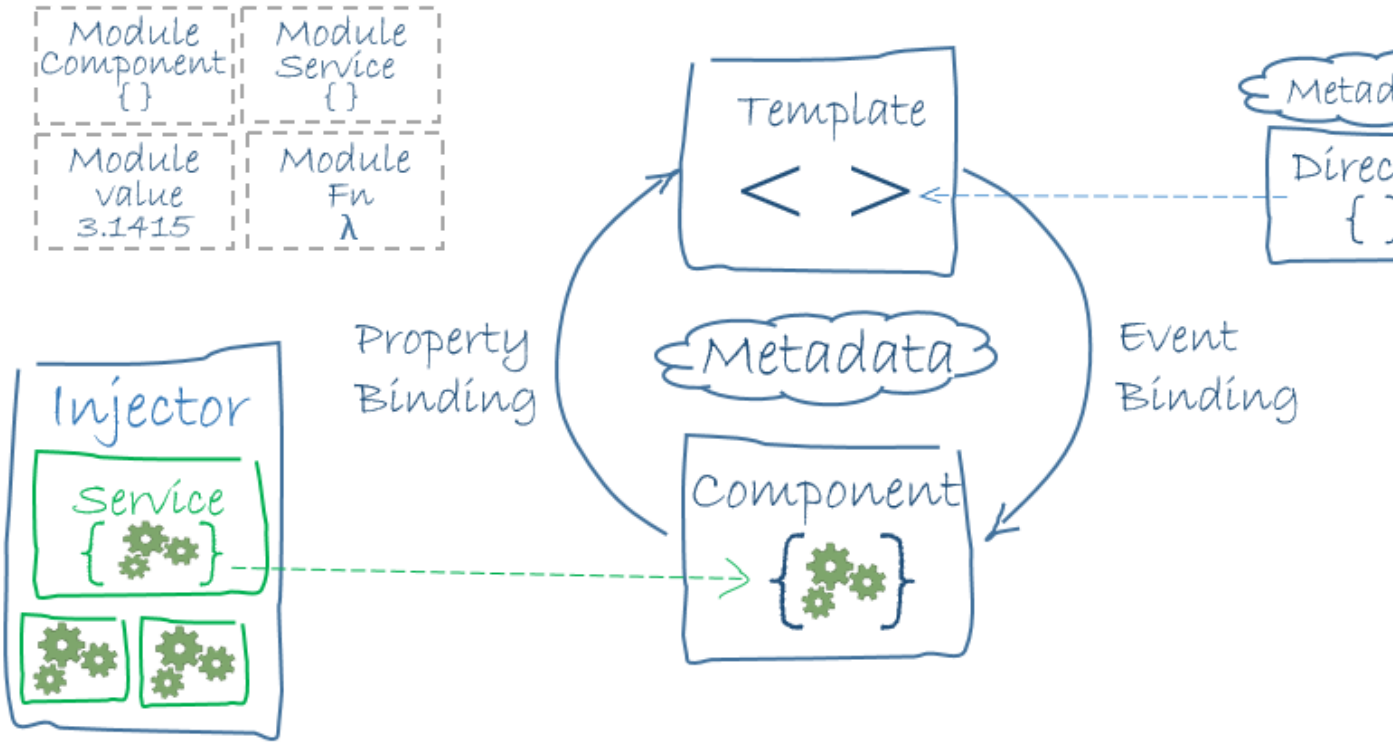
```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

BrowserModule	It is required for Angular application to load the scripts according to the browser version and type.
NgModule	- It is the base for all modules in Angular. - It provides members that required for module.
declarations[]	- It is a collection of all **components** and **pipes** that are used for application. - If any component or pipe is not configured in declarations then it is not accessible in application.
imports[]	- It is a collection all **modules** in

	application.
providers[]	- It is a collection of **services** used by application.
bootstrap[]	- It comprises of components that you want on application start up.



Angular Application Architecture


Module
Component
{}
Module
Service
{}
Module
value
3.1415
Module
Fn
λ
Template
< >
Metad
Direc
{
Property
Binding
Metadata
Event
Binding
Injector
Service
Component