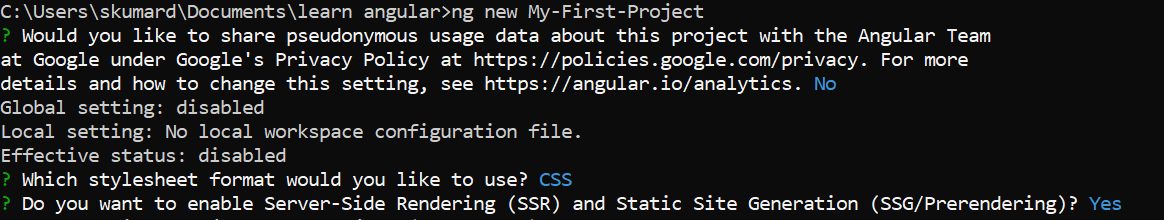
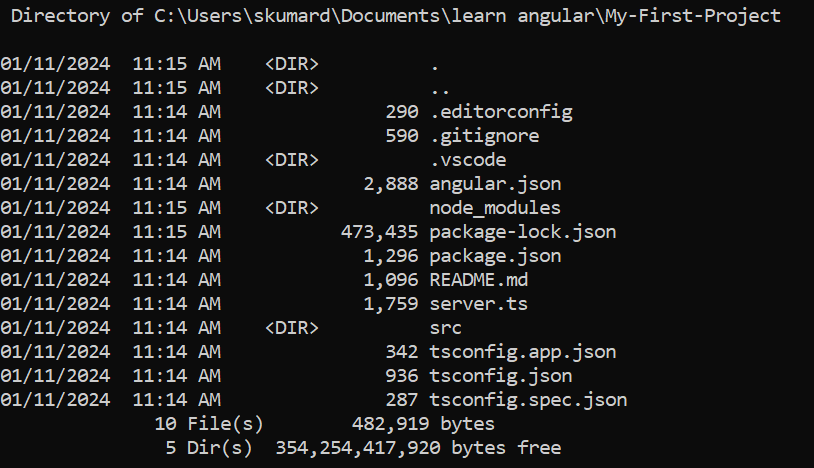
**Creating new angular project**

Use command: *ng new <Project-Name>*



Angular project directory str:



By default angular projects are standalone. So use the below command to create a no standalone app:

*ng new --no-standalone <project-name>*

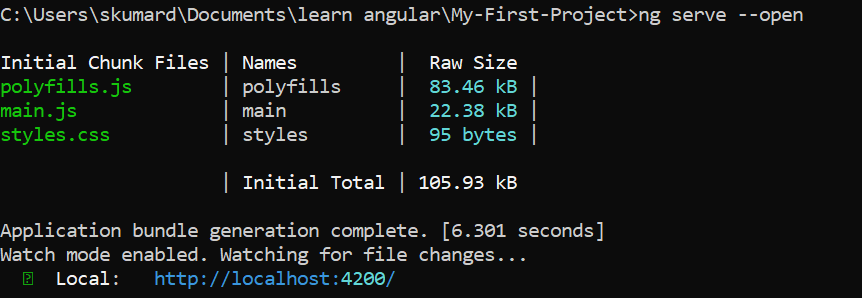
By default strict typescript is enabled. To create a project with strict disabled:

*ng new --no-standalone <project-name> --no-strict*

**Running angular project**:

*ng serve* : this will make the app running at localhost:4200

*ng serve –open* : this will do the same as above in addition it will also open the default browser with the root app url.



**To use bootstrap as css:**

*npm install --save* [*bootstrap@5.3*](mailto:bootstrap@5.3)

then make changes in angular.json



If you plan to have multiple applications in the workspace, you can create an empty workspace by using the --no-create-application option. You can then use ng generate application to create an initial application. This allows a workspace name different from the initial app name, and ensures that all applications reside in the /projects subfolder, matching the structure of the configuration file.

**Generate component:**

Use : *ng generate component <component-name>*

Or: *ng g c <component-name>*

🡪This will create a directory under app folder with the name as <component-name>.

🡪It will also add 4 files to it:

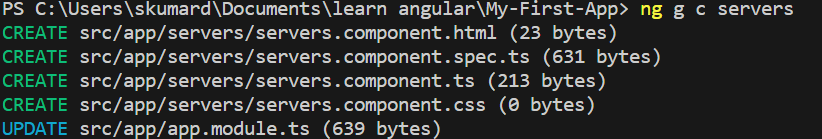
<component-name>.component.html

<component-name>.component.ts

<component-name>.component.css

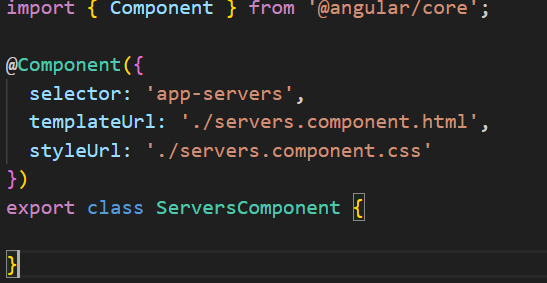
<component-name>.component.spec.ts

🡪It will also update the app.module.ts file to include it

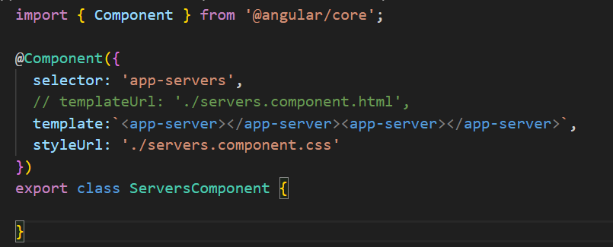


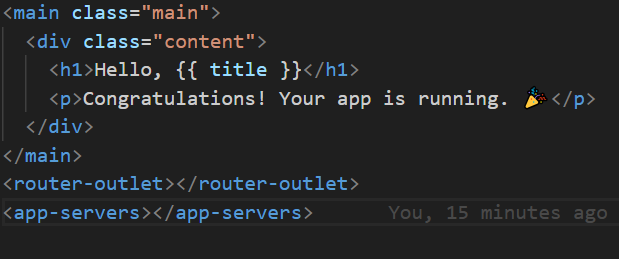
**To use a component**:

See the <component-name>.component.ts file



A component can be used any number of times using its selector in either a template html file pointed by templateUrl or inline with template.

With a template html file: with template:



But we can have either templateUrl or template but not both.

Same is true for styleUrls and styles.

Selectors can be declared and used in three ways:

1. element style : selector: 'app-servers'

to use this in template html use: <app-servers></app-servers>

1. attribute style: selector: '[app-servers]'

to use this in template html use: <div app-servers></div>

1. class style: selector: '.app-servers'

to use this in template html use: <div class=”app-servers”></div>

**Databinding**

Communication between ts and html.

String Interpolation.

Property Binding.

Event Binding.

Two-way binding

<button

    class="btn btn-primary"

    [disabled] = "!allowedNewServer" ---property binding

    (click)="onCreateServer()" ---event binding

    >Add Server</button>

<p [innerText]="serverCreationStatus"></p>---property binding

<p>{{serverCreationStatus}}</p> --string Interpolation

The values being provided in case of each binding must be defined inline or in the ts file of the component.

For two way binding :

Include FormsModule in the imports array of app.module.ts

  imports: [

    FormsModule //required for two way binding

  ],

And in the html template use ngModel directive . it also helps to prepopulate a form.

<input

    type="text"

    class="form-control"

    [(ngModel)]="serverName">

The two way binding is a reactive approach to angular.

**Directives**

Directives are classes that add additional behavior to elements in your Angular applications

The different types of Angular directives are as follows:

|  |  |
| --- | --- |
| DIRECTIVE TYPES | DETAILS |
| Components | Used with a template. This type of directive is the most common directive type. |
| Attribute directives | Change the appearance or behavior of an element, component, or another directive. |
| Structural directives | Change the DOM layout by adding and removing DOM elements. |

ngIf is an structural directive.

<p \*ngIf="serverCreated; else noServer;">{{serverCreationStatus}}</p>

<ng-template #noServer>

    <p>No Server was created</p>

</ng-template>

ngStyle is an attribute directive.

<p [ngStyle]="{backgroundColor: getColor()}">server with id {{ serverId}} is {{ serverStatus}}.</p>

ngClass is an attribute directive.

<p

    [ngStyle]="{backgroundColor: getColor()}"

    [ngClass]="{online: serverStatus==='online'}">server with id {{ serverId}} is {{ serverStatus}}.</p>