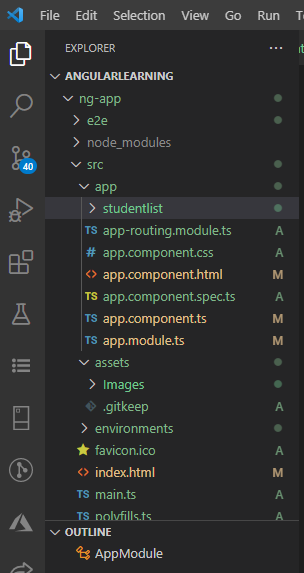
**Create a studentlist Component**

**Ng g c studentlist**



A studentlist component is created in a separate studentlist folder

import { Component, OnInit } from '@angular/core';

@Component({

selector: 'app-studentlist',

templateUrl: './studentlist.component.html',

styleUrls: ['./studentlist.component.css']

})

export class StudentlistComponent {

constructor() { }

students: any[] = [

{

ID: 'std101', FirstName: 'Preety', LastName: 'Tiwary',

Branch: 'CSE', DOB: '29/02/1988', Gender: 'Female'

},

{

ID: 'std102', FirstName: 'Anurag', LastName: 'Mohanty',

Branch: 'ETC', DOB: '23/05/1989', Gender: 'Male'

},

{

ID: 'std103', FirstName: 'Priyanka', LastName: 'Dewangan',

Branch: 'CSE', DOB: '24/07/1992', Gender: 'Female'

},

{

ID: 'std104', FirstName: 'Hina', LastName: 'Sharma',

Branch: 'ETC', DOB: '19/08/1990', Gender: 'Female'

},

{

ID: 'std105', FirstName: 'Sambit', LastName: 'Satapathy',

Branch: 'CSE', DOB: '12/94/1991', Gender: 'Male'

}

];

ngOnInit(): void {

}

}

**studentlist.component.html**

<html>

<table>

  <thead>

      <tr>

          <th>ID</th>

          <th>First Name</th>

          <th>Last Name</th>

          <th>Branch</th>

          <th>DOB</th>

          <th>Gender</th>

      </tr>

  </thead>

  <tbody>

      <tr \*ngFor='let student of students'>

          <td>{{student.ID}}</td>

          <td>{{student.FirstName}}</td>

          <td>{{student.LastName}}</td>

          <td>{{student.Branch}}</td>

          <td>{{student.DOB}}</td>

          <td>{{student.Gender}}</td>

      </tr>

      <tr \*ngIf="!students || students.length==0">

          <td colspan="7">

              No Students to display

          </td>

      </tr>

  </tbody>

</table>

</html>

Copy the selector of studenlist for example

  selector: 'app-studentlist',

**app-studentlist is the selector value put this in root component ‘s Template**

import { Component } from '@angular/core';

import {StudentlistComponent} from  './studentlist/studentlist.component';

@Component({

  selector: 'app-root',

  template: `<div>

  <app-studentlist></app-studentlist>

**Register the studenlistComponent in ap-.modules.ts**

**Inside declarations array and import studentlist**

**On top of the @NgModule**

**import { StudentlistComponent } from './studentlist/studentlist.component';**

import { StudentlistComponent } from './studentlist/studentlist.component';

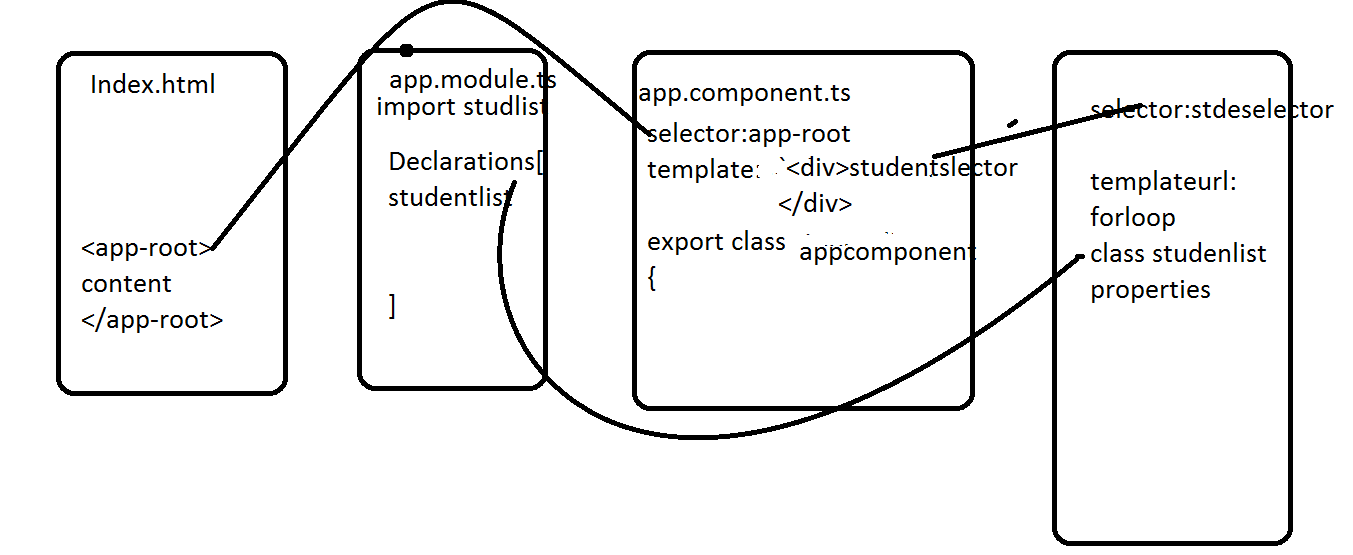
@NgModule({

  declarations: [

    AppComponent,

    StudentlistComponent

  ],

****

NgSwitch is a built-in Angular 11 template directive, and it works the same way as the JavaScript's switch statement does. ... This directive helps in showing or hiding the templates based on the conditional statement

**NgSwitch Directive**

The [ngSwitch] directive on a container specifies an expression to match against. The expressions to match are provided by ngSwitchCase directives on views within the container.

* Every view that matches are rendered.
* If there are no matches, a view with the ngSwitchDefault directive is rendered.
* Elements within the [NgSwitch] statement but outside of any NgSwitchCase or ngSwitchDefault directive are preserved at the location.

**Syntax**

Define a container element for the directive, and specify the switch expression to match against as an attribute:

**<container-element [ngSwitch]="switch\_expression">**

Within the container, \*ngSwitchCase statements specify the match expressions as attributes. Include \*ngSwitchDefault as the final case.

<container-element [ngSwitch]="switch\_expression">

<some-element \*ngSwitchCase="match\_expression\_1">...</some-element>

...

<some-element \*ngSwitchDefault>...</some-element>

</container-element>

Sample example:

<!-- ngSwitch and ngSwitchCase - enum example -->

<b> ngSwitch and ngSwitchCase - enum example</b>

<div [ngSwitch]="day">

<b \*ngSwitchCase="days.SUNDAY"> SUNDAY</b>

<b \*ngSwitchCase="days.MONDAY"> MONDAY</b>

<b \*ngSwitchCase="days.TUESDAY">TUESDAY</b>

<b \*ngSwitchCase="days.WEDNESDAY">WEDNESDAY</b>

<b \*ngSwitchCase="days.THURSDAY">THURSDAY</b>

<b \*ngSwitchCase="days.FRIDAY">FRIDAY</b>

<b \*ngSwitchCase="days.SATURDAY">SATURDAY</b>

<b \*ngSwitchDefault>No Days</b>

</div>

# A Step-by-Step Guide to NgSwitch

Now that you know what the NgSwitch directive does, let’s look at a simple demo explaining the same.

UseCase - To display a message according to the day of the week.

Step 1 - Create a component in your Angular App folder using the command.

ng g c <name\_of\_component>

The component that was created is called newcomp.

Step2 - In the newcomp.ts file, create a day property that stores the current day of the week.

export class NewcompComponent implements OnInit {

  constructor() { }

  public day = new Date().getDay()

    ngOnInit(): void {

  }

}

The getDay() method returns a digit corresponding to the days of the week. It returns 0 for Sunday, 1 for Monday, and so on.

Step 3 - In the newcomp.html file, use the NgSwitch directive to display messages according to the value stored in the day property.

<h1>Hey! Good Morning </h1>

<div [NgSwitch] = "day">

<div \*NgSwitchCase = 0>Today is Sunday! You deserve a break</div>

<div \*NgSwitchCase = '1'><h3>Today is Monday, Hope you have productive Week</h3></div>

<div \*NgSwitchCase = '2'><h3>Today is Tuesday, Smash your Goals</h3></div>

<div \*NgSwitchCase = '3'><h3>Today is Wednesday, You're already half way through</h3></div>

<div \*NgSwitchCase = '4'><h3>Today is Thursday, It's like Friday Lite</h3></div>

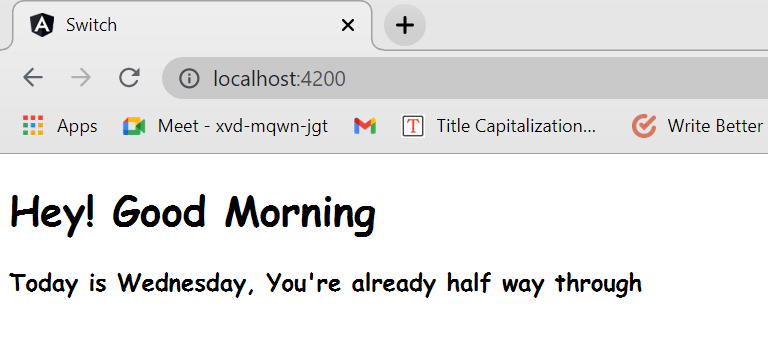
<div \*NgSwitchCase = '5'><h3>Today is Friday, TGIF!!!</h3></div>

<div \*NgSwitchCase = '6'><h3>Today is Saturday, Let's Party!! :) </h3></div>

<div \*NgSwitchDefault>Uh oh! Somethings wrong</div>

</div>

Step 4 - Run your app using the ng serve command. The output will look like this.

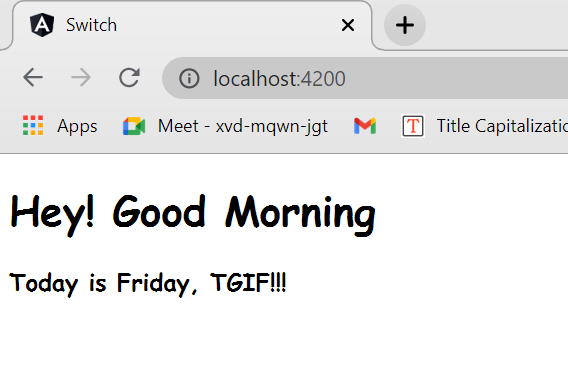


Since this article was written on a Wednesday, accordingly the message is being displayed.

Step 5 - If you wish to customize the message being displayed, you can create another property with a hard-coded value.

public day2 = 5

The output for the same would be.

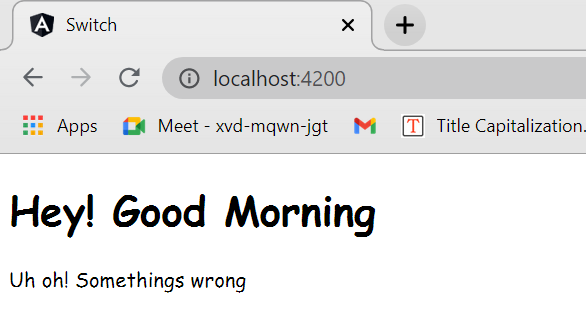




Step 6 - Let’s consider a scenario where the expression to match is invalid. In this case, it must provide a value above 7 for the day2 property.

  public day2 = 8

The ngSwitchDefault directive is rendered. The output will look like this.



Example-2 for with switch example

