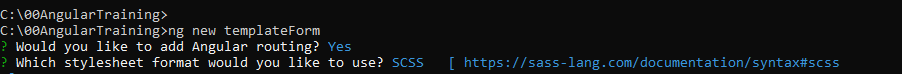
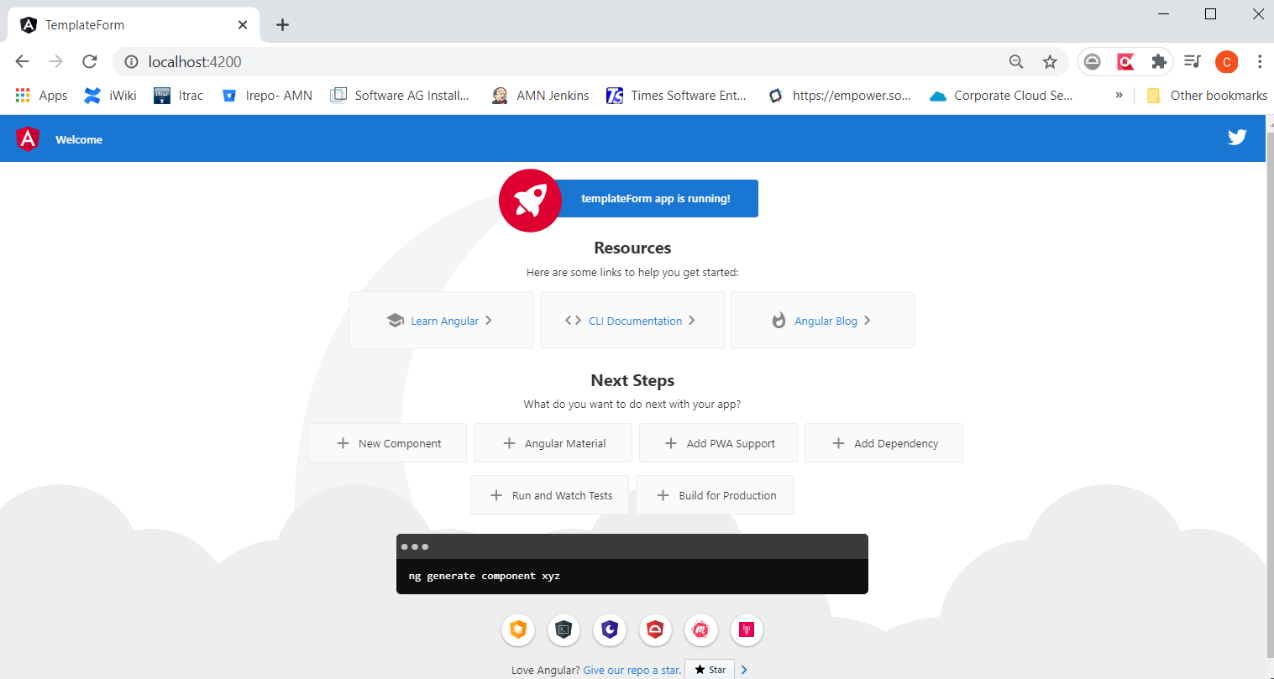
Preparation

1. Create New Angular Project
   1. ng new <project name>



* 1. ng serve



1. Add Bootstrap – add below into ./src/styles.css

@import url('https://unpkg.com/bootstrap@3.3.7/dist/css/bootstrap.min.css')

Exercise

1. Create a new “testIf” directive
2. Add below codes into test-if.directive.ts

  @Input()

  set appTestIf(condition) {

    condition ? this.\_viewContainer.createEmbeddedView(this.templateRef) : this.\_viewContainer.clear();

  }

  constructor(private \_viewContainer: ViewContainerRef,

    private templateRef: TemplateRef<any>) {

  }

1. Replace app.component.html with below

<h2>Structural Directive</h2>

<br/>

<button (click)="flag = !flag">change flag</button>

<div \*ngIf="flag">

  ngif show {{flag}}

</div>

<ng-template  [ngIf]="flag">

  <div>

    ngif2 show {{flag}}

  </div>

</ng-template>

<div \*appTestIf="flag">

  appTestIf show {{flag}}

</div>

<ng-template  [appTestIf]="flag">

  <div>

    appTestIf2 show {{flag}}

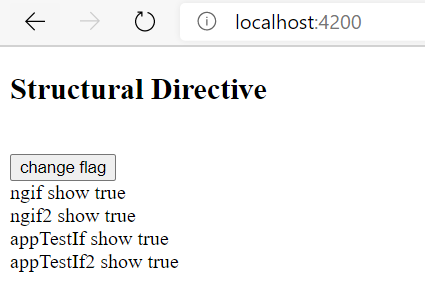
  </div>

</ng-template>

1. Add below code into app.component.ts

flag = false;

1. Test the application, click on the button to display or not display the texts



1. Create a new “delay” directive
2. Add below codes into delay.directive.ts

  constructor(private templateRef: TemplateRef<any>,

    private viewContainer: ViewContainerRef) { }

  @Input() set appDelay(delay: number) {

    this.viewContainer.clear();

    setTimeout(() => {

      this.viewContainer.createEmbeddedView(this.templateRef);

    }, delay);

  }

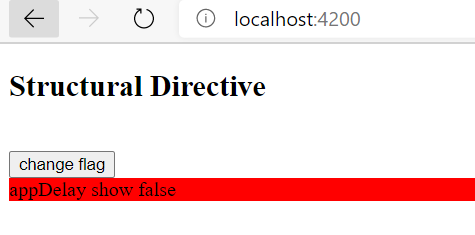
1. Add below codes into app.component.html

<div \*appDelay="3000" style="background-color: red;">

  appDelay show {{flag}}

</div>

1. Test the application, wait for 3 seconds to see the text



1. Create a new “loop” directive
2. Add below codes into loop.directive.ts

  constructor(private templateRef: TemplateRef<any>,

    private viewContainer: ViewContainerRef) { }

  @Input() set appLoop(num: number) {

    for (var i = 0; i < num; i++) {

      this.viewContainer.createEmbeddedView(this.templateRef);

    }

  }

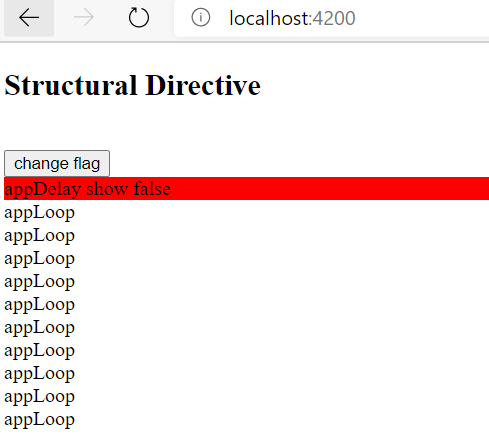
1. Add below codes into app.component.html

<div \*appLoop="10">

  appLoop

</div>

1. Test the application, you will see the text repeated 10 times



1. Create a new “objectNgFor” directive
2. Add below codes into object-ng-for.directive.ts

  @Input() set appObjectNgForFrom(inputObj: any) {

    this.viewContainerRef.clear();

    Object.keys(inputObj).forEach((propertyName: string, index: number) => {

      console.log(propertyName, index);

      this.viewContainerRef.createEmbeddedView(this.templateRef, {

        // default value that will be used if an attribute is not assigned one

        $implicit: propertyName,

        index

      });

    });

  };

  constructor(

    private templateRef: TemplateRef<any>,

    private viewContainerRef: ViewContainerRef

  ) { }

1. Add below codes into app.component.html

<div \*appObjectNgFor="let propertyName from myObject; let index=index">

  appObjectNgFor [{{index}}] {{propertyName}}: {{myObject[propertyName]}}

</div>

1. Add below codes into app.component.html

  myObject =

    {

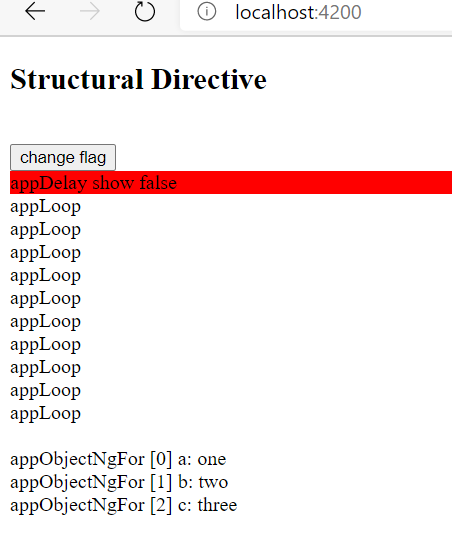
      a: 'one',

      b: 'two',

      c: 'three'

    }

1. Test the application, you will able to see the object’s items been loop thru and shown as below



Additional Exercise

1. Create a directive that show the element when condition is false
2. Create a directive that loop thru below object but skip the property with value greater than 10

Obj = {

a : 1,

b : 11,

c : 111,

d : 9,

e : 5,

f : 200

}