Angular 4 - Templates

**Angular 4** uses the **<ng-template>** as the tag instead of **<template>**which is used in Angular2. The reason Angular 4 changed **<template>** to **<ng-template>** is because there is a name conflict between the **<template>** tag and the html **<template>** standard tag. It will deprecate completely going ahead. This is one of the major changes in Angular 4.

Let us now use the template along with the **if else** condition and see the output.

app.component.html

<!--The content below is only a placeholder and can be replaced.-->

<div style = "text-align:center">

<h1>

Welcome to {{title}}.

</h1>

</div>

<div> Months :

<select (change) = "changemonths($event)" name = "month">

<option \*ngFor = "let i of months">{{i}}</option>

</select>

</div>

<br/>

<div>

<span \*ngIf = "isavailable;then condition1 else condition2">Condition is valid.</span>

<ng-template #condition1>Condition is valid from template</ng-template>

<ng-template #condition2>Condition is invalid from template</ng-template>

</div>

<button (click) = "myClickFunction($event)">Click Me</button>

For the Span tag, we have added the **if** statement with the **else** condition and will call template condition1, else condition2.

The templates are to be called as follows −

<ng-template #condition1>Condition is valid from template</ng-template>

<ng-template #condition2>Condition is invalid from template</ng-template>

If the condition is true, then the condition1 template is called, otherwise condition2.

app.component.ts

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

@Component({

selector: 'app-root',

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

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

})

export class AppComponent {

title = 'Angular 4 Project!';

//array of months.

months = ["January", "February", "March", "April",

"May", "June", "July", "August", "September",

"October", "November", "December"];

isavailable = false;

myClickFunction(event) {

this.isavailable = false;

}

changemonths(event) {

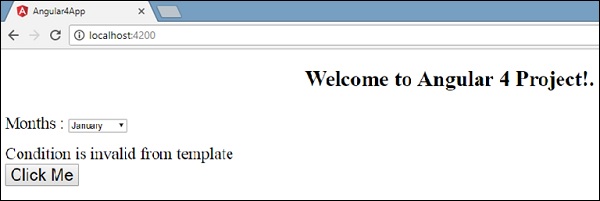
alert("Changed month from the Dropdown");

console.log(event);

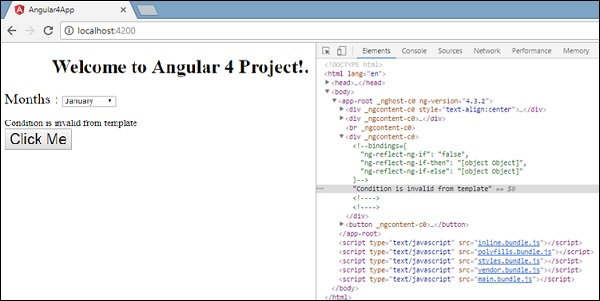
}

}

The output in the browser is as follows −



The variable **isavailable** is false so the condition2 template is printed. If you click the button, the respective template will be called. If you inspect the browser, you will see that you never get the span tag in the dom. The following example will help you understand the same.



If you inspect the browser, you will see that the dom does not have the span tag. It has the **Condition is invalid from template** in the dom.

The following line of code in html will help us get the span tag in the dom.

<!--The content below is only a placeholder and can be replaced.-->

<div style = "text-align:center">

<h1>

Welcome to {{title}}.

</h1>

</div>

<div> Months :

<select (change) = "changemonths($event)" name = "month">

<option \*ngFor = "let i of months">{{i}}</option>

</select>

</div>

<br/>

<div>

<span \*ngIf = "isavailable; else condition2">Condition is valid.</span>

<ng-template #condition1>Condition is valid from template</ng-template>

<ng-template #condition2>Condition is invalid from template</ng-template>

</div>

<button (click)="myClickFunction($event)">Click Me</button>

If we remove the then condition, we get the **“Condition is valid”** message in the browser and the span tag is also available in the dom. For example, in **app.component.ts**, we have made the **isavailable** variable as true.

