**What is Event Binding in Angular?**

When a user interacts with an application in the form of a keyboard movement, button click, mouse over, selecting from a drop-down list, typing in a textbox, etc. it generates an event. These events need to be handled to perform some kind of action. This is where event binding comes into the picture and in Angular Application, we can use event binding to get notified when these events occur.



**Angular Event Binding Example:**

Let us understand Angular Event Binding with an example. Please modify the **app.component.ts** file as shown below.

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

@Component**({**

selector: 'app-root',

template: `<div>

<button (click)="onClick()">Click Me </button>

</div>`

**})**

**export** **class** AppComponent **{**

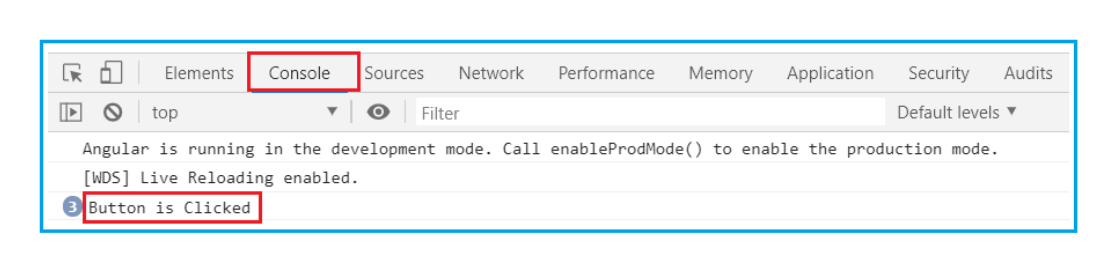
onClick**()**: **void** **{**

console.log**(**'Button is Clicked'**)**;

**}**

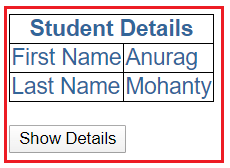
**}**

Now, run the application and launch the browser developer tools by pressing the F12 key. Once you open the browser developer tools click on the Console tab as shown in the image below. Notice every time you click the button, ‘Button is Clicked’ message is displayed on the console.

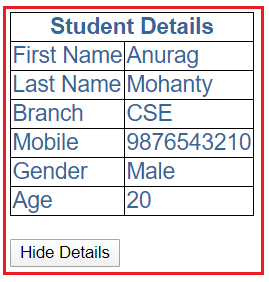


**Another Example:**

When the page loads for the first time, we want to display only the First Name and Last Name of the student. We also display the “Show Details” button as shown in the below image.



When the user clicks on the **“Show Details”** button, we want to display the “**Gender**“, “**Age**“, “**Mobile**”, and “**Branch**” as well. The text on the button should be changed to **“Hide Details”**as shown in the below image and when the user clicks on the **“Hide Details”** button, then the “**Gender**“, “**Age**“, “**Mobile**”, and “**Branch**”should be hidden and the button text should be changed to **“Show Details”**.



We can achieve this very easily in angular with the help of event binding. Here we will make use of one of the angular directives i.e. “**ngIf**“.

**Modify app.component.ts file:**

Notice we have introduced “ShowDetails” boolean property. The default value is false, so when the page loads for the first time, we will have “Gender”, “Age”, “Mobile”, and “Branch” hidden. We also have a method, ToggleDetails(), which will toggle the value of ShowDetails.

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

@Component**({**

selector: 'app-root',

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

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

**})**

**export** **class** AppComponent **{**

ColumnSpan: **number** = 2;

FirstName: **string** = 'Anurag';

LastName: **string** = 'Mohanty';

Branch: **string** = 'CSE';

Mobile: **number** = 9876543210

Gender: **string** = 'Male';

Age: **number** = 20;

ShowDetails: **boolean** = **false**;

ToggleDetails**()**: **void** **{**

this.ShowDetails = !this.ShowDetails;

**}**

**}**

**Modify app.component.html file:**

Notice the click event of the button element is bounded to ToggleDetails() method. To dynamically change the text on the button, we are using a ternary operator:

**{{ShowDetails ? ‘Hide’ : ‘Show’}} Details**

We used ngIf structural directive on “Gender”, “Branch”, “Mobile” and “Age” <tr> elements. The \* prefix before a directive indicates, it is a structural directive. Besides ngIf, there are other structural directives which we will discuss in our upcoming articles.

The ngIf directive conditionally adds or removes content from the DOM based on whether or not an expression is true or false. If “**ShowDetails**” is true, “Gender”, “Branch”, “Mobile” and “Age” <tr> elements are added to the DOM, else removed.

**<**table**>**

**<**thead**>**

**<**tr**>**

**<**th attr.colspan="{{ColumnSpan}}"**>**

Student Details

**<**/th**>**

**<**/tr**>**

**<**/thead**>**

**<**tbody**>**

**<**tr**>**

**<**td**>**First Name**<**/td**>**

**<**td**>{{**FirstName**}}<**/td**>**

**<**/tr**>**

**<**tr**>**

**<**td**>**Last Name**<**/td**>**

**<**td**>{{**LastName**}}<**/td**>**

**<**/tr**>**

**<**tr \*ngIf='ShowDetails'**>**

**<**td**>**Branch**<**/td**>**

**<**td**>{{**Branch**}}<**/td**>**

**<**/tr**>**

**<**tr \*ngIf='ShowDetails'**>**

**<**td**>**Mobile**<**/td**>**

**<**td**>{{**Mobile**}}<**/td**>**

**<**/tr**>**

**<**tr \*ngIf='ShowDetails'**>**

**<**td**>**Gender**<**/td**>**

**<**td**>{{**Gender**}}<**/td**>**

**<**/tr**>**

**<**tr \*ngIf='ShowDetails'**>**

**<**td**>**Age**<**/td**>**

**<**td**>{{**Age**}}<**/td**>**

**<**/tr**>**

**<**/tbody**>**

**<**/table**>**

**<**br/**>**

**<**button **(**click**)**='ToggleDetails()'**>**

**{{**ShowDetails ? 'Hide' : 'Show'**}}** Details

**<**/button**>**

**Modify app.component.css file:**

Modify the app.component.css file as shown below.

table **{**

color: *#369*;

font-family: Arial, Helvetica, sans-serif;

font-size:large;

border-collapse: collapse;

**}**

td **{**

border: 1px solid black;

**}**

thead**{**

border: 1px solid black;

**}**

Now run the application and you will see everything is working as expected as per our requirement.

Example for Event Binding with parameter

<!-- event is passed to function -->

<input (keyup)="onKeyUp($event)">

<p>{{text}}</p>

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'Demo';

  text = '';

  onKeyUp(element:any) {

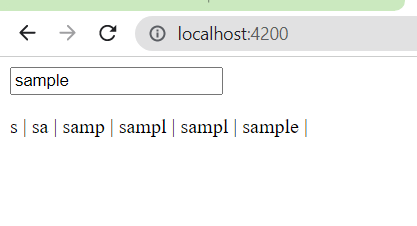
    // Appending the updated value

    // to the variable

    this.text += element.target.value + ' | ';

  }

}



Bind Event without ngModel

<input type="text" (input)="userName=$event.target.value"><br/>

{{userName}}