Data binding

# Introduction to Data binding

Binding coordinates the communication between the component class and template and often involves in passing data.

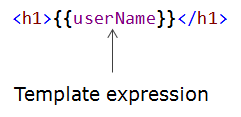
You can provide values from the class to the template for display and the template raises events to pass the user actions or user entered values back to the class.

Angular provides the below types of binding

* Interpolation
* Property binding
* Event binding
* Two way binding

Interpolation is a one way binding technique. It is recognized using {{}} (double curly braces).

The information that is present in between the curly braces is called template expression. Angular evaluates the expression converts the result into a string and assigns it to an HTML element or attribute property.



Interpolation can be used to

* bind a value from the component's class to the template
* perform operations such as concatenation, calculations or method invocations etc.

Example

1. <h1>{{'Welcome'+ userName}}</h1> *<!--concatenation-->*
2. <p>{{20\*10+1}}</p> *<!--calculation-->*
3. <div>{{'Balance is' + getBalance()}}</div> *<!--method invocation-->*

Angular prevents Cross-site scripting (XSS) that enables attackers to inject malicious code into web pages. It treats all values as untrusted by default. When a value is inserted into the DOM from a template, via property, attribute, style, class binding, or interpolation, Angular sanitizes and escapes untrusted values.

Interpolated content is always escaped. This means the HTML will not be interpreted and the browser will display the data in the element's text content.

1. <p>{{<b>Hello</b>}}</p> *<!--This will display the output as <b>Hello</b> instead of showing Hello in bold -->*

Property binding is also a one way binding technique. It helps you to set the property of an HTML element to the value of a template expression.

Syntax:

1. <element [binding target]='template expression'/>

Here binding target is the property of the html element to be set. It should be enclosed with in []. The template expression is the binding source (property of component class) which should be enclosed in ' '.

Example:

1. <img [src]='card.imageUrl'/>

You can bind values to the elements either using interpolation or property binding. In general property binding is preferred over interpolation. However if you need to include a template expression as a part of a larger expression you may need to use interpolation.

Example:

1. <img src='http://amigowallet.org/{{card.imageUrl}}'/>

Event binding is used to send information from a template to the component's class based on some user actions like click of a button.

Syntax:

1. <element (target event)='Template statement'/>

Here target event is the name of the event on which information should be sent to the component's class. It should be enclosed with in (). Template statement is often the name of the component class method to be invoked on the occurrence of the event.

Example:

1. <button (click) ='getTransactions()'/>

In the above code getTransactions() is a method of the associated component class which will be invoked on click of the button.

You will use interpolation and event binding to display the products belonging to a selected category in QuickKart.

# Two way binding

Two way binding helps to bind a property from the component class to an input element in the template and update the property of the class if the bound value is modified by the user.

Syntax:

1. <input [(ngModel)]='template expression'/>

Here [] is used to bind the property value of the component to an input element and () is used to send notification of the user modified data back to the class.

Example:

1. <input [(ngModel)] ='emailid'/>

Here the value of emailid will be bound to the control. If the user modifies the the emailid it will be notified to the component class.

# Two way binding in Angular - Demo

Highlights:

* To learn two way data binding
* To learn about ngModel

Demosteps:

In order to learn two way data binding in Angular, try out the below demo steps.

In this demo you will read the firstname and lastname of the user and concatenate and display it as fullname using two way binding.

**Step 1:**

Create a new app TryOutApp using the following cli command

1. ng new TryOutApp

Open the TryOutApp folder in Visual studio.

**Step 2:**

Create a new component in TryOutApp using the following cli command

1. ng generate component twoWayBinding

**Step 3:**

Open two-way-binding.component.ts in TryOutApp-->src-->app-->two-way-binding folder and declare the two variables firstName and lastName of type string to store the first name and last name entered by the user in line 10 and 11.

1. import { Component, OnInit } from '@angular/core';
2. @Component({
3. selector: 'app-two-way-binding',
4. templateUrl: './two-way-binding.component.html',
5. styleUrls: ['./two-way-binding.component.css']
6. })
7. export class TwoWayBindingComponent implements OnInit {
8. firstName: string;
9. lastName: string;
10. constructor() { }
11. ngOnInit() {
12. }
13. }

**Step 4:**

Open two-way-binding.component.html in TryOutApp-->src-->app-->two-way-binding folder.

Replace the code written in two-way-binding.component.html with the below code.

1. <h3 style="text-align:center;">Two Way Data Binding</h3>
2. <br />
3. <div class="row">
4. <div class="col-md-4" style="text-align:right">
5. </div>
6. <div class="col-md-3" style="margin-left:55px; background-color:beige">
7. <br />
8. <label>First Name</label>&nbsp;
9. <input type="text" [(ngModel)]="firstName" />
10. <br />
11. <br />
12. <label>Last Name</label>&nbsp;
13. <input type="text" [(ngModel)]="lastName" />
14. <br />
15. <br />
16. <label>Full Name:</label> {{firstName}} {{lastName}}
17. </div>
18. <div class="col-md-4" style="text-align:right">
19. </div>
20. </div>

Observe that here ngModel is used. In angular ngModel directive is used for two way binding(view to component and vice versa).

Here the ngModel is assigned firstName which is declared in TwoWayBindingComponent class to store the first name entered by user.

Similarly the ngModel is assigned lastName which is declared in TwoWayBindingComponent class to store the last name entered by user.

ngModel is wrapped in parenthesis () and square brackets [] i.e. [(ngModel)]. This is called **banana in the box** model.

Recollect from the previous demo event data binding is specified by parenthesis () and property data binding is specified by square brackets []. Two way data binding makes use of both event data binding and property data binding.

Whenever the firstName and lastName property gets changed the view is updated accordingly and when the view changes i.e. when user enters first name and last name in the web page,  firstName and lastName property defined in TwoWayBindingComponent gets updated.

Also interpolation is used to display the live preview of the firstName and lastName property.

**Step 5:**

To include ngModel you should import FormModel in app.module.ts file.

Include the code written below in app.module.ts in line 4 and 16.

1. import { BrowserModule } from '@angular/platform-browser';
2. import { NgModule } from '@angular/core';
3. import { FormsModule } from '@angular/forms';
4. import { AppComponent } from './app.component';
5. import { TwoWayBindingComponent } from './two-way-binding/two-way-binding.component';
6. @NgModule({
7. declarations: [
8. AppComponent,
9. TwoWayBindingComponent
10. ],
11. imports: [
12. BrowserModule,
13. FormsModule
14. ],
15. providers: [],
16. bootstrap: [AppComponent]
17. })
18. export class AppModule { }

**Step 6:**

Open app.component.html and add the following line 4 to it, so that the template of TwoWayBindingComponent can be rendered in the view.

Note that here TwoWayBindingComponent is the child component and AppComponent is parent component.

1. <div>
2. <h1 style="text-align:center;">Try out demo</h1>
3. <br>
4. <app-two-way-binding></app-two-way-binding>
5. </div>

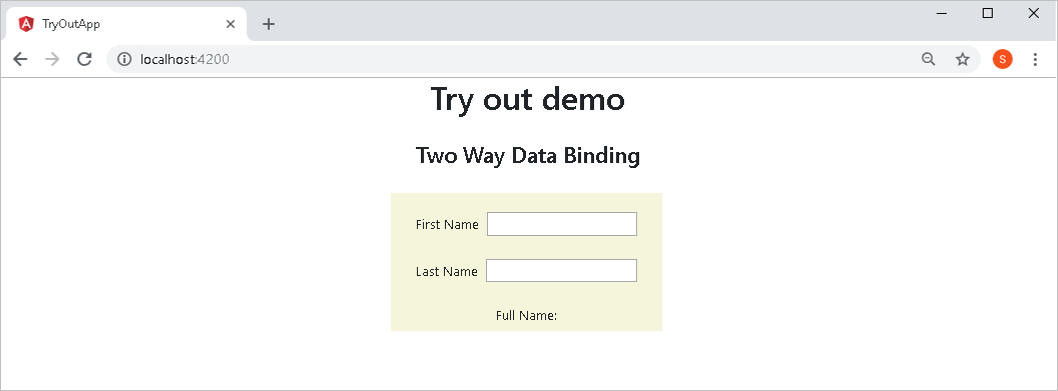
**Note**: Install bootstrap and font-awesome as shown earlier in the course.

**Step 7:**

Execute your app by using the following cli command

1. ng serve -o

In browser, you can see the page will be loaded as shown below.



Enter the First Name and Last Name, and see Full Name in the live preview as shown below.

