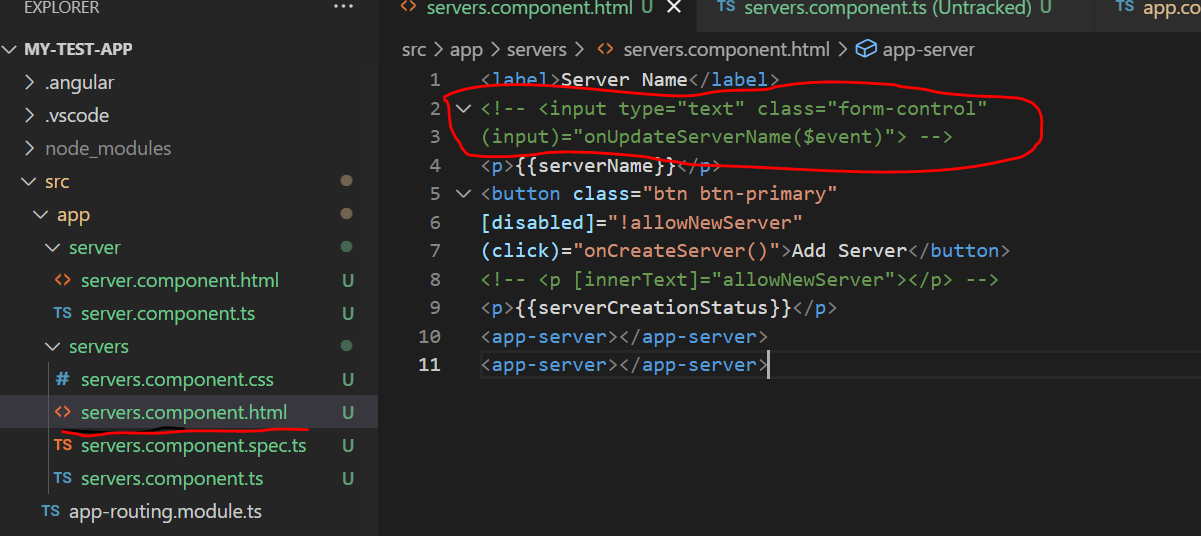
1. FormsModule is required for Two-Way Databinding
2. Until now, we have seen event binding and property binding.
3. Now lets combine both to two-way binding
4. We learnt that we can get event data like our previous example:
5. Giving an input
6. For the input call a method: onUpdateServerName($event)
7. onUpdateServerName will get input value entered from $ event and populate serverName property
8. Later serverName property is rendered in the html
9. **Another Approach:**

* As first step, lets comment the below lines of servers.component.html file



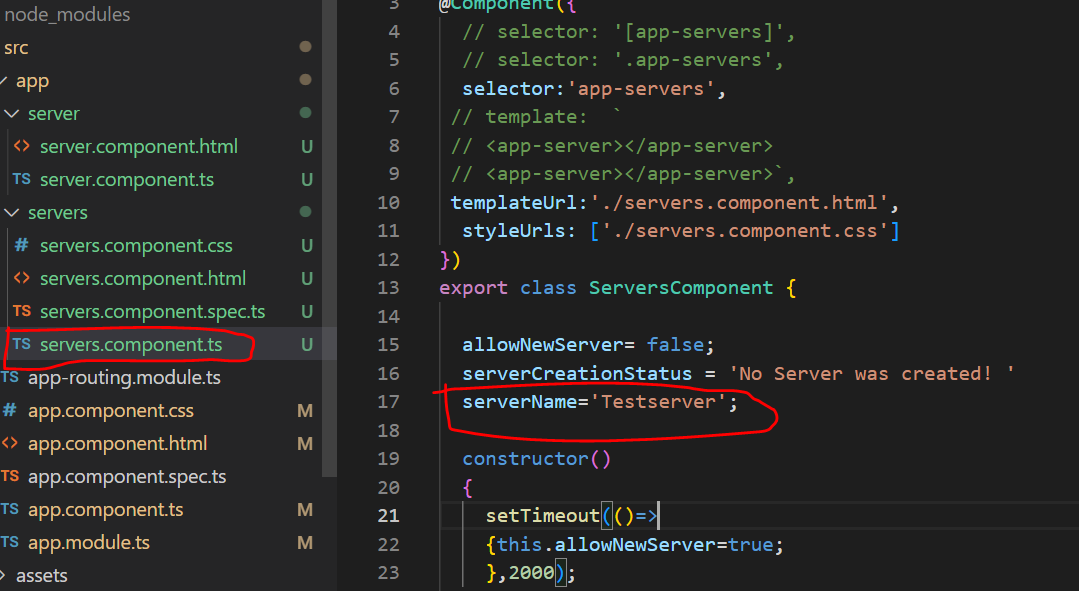
**Data Binding syntax:**

* Since Two way binding is binding both property and event binding, we use it by combining the syntaxes:
* We have square brackets and within it we have parenthesis. Within parenthesis we use special directive:ngModel. This ngModel is set equal to some property defined in our typescript code.

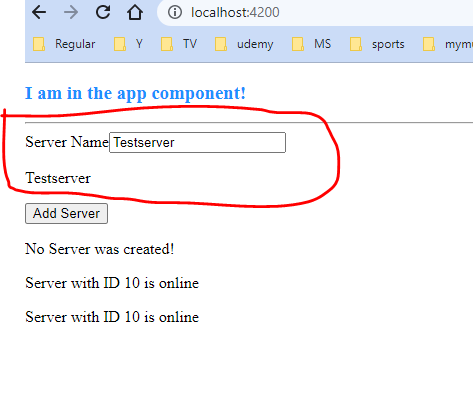
<input type="text" class="form-control"

[(ngModel)]="serverName">

* Above code will do the following:
* It gets triggered on the input event.
* And updates the value of server name in our component automatically
* On the other hand, since it is two way binding – it will also update the value of the input element., if we change server name somewhere else.
* Now we can demonstrate this by going to ServersComponent(servers.component.ts) file and change servername with some default name, which was earlier a empty string



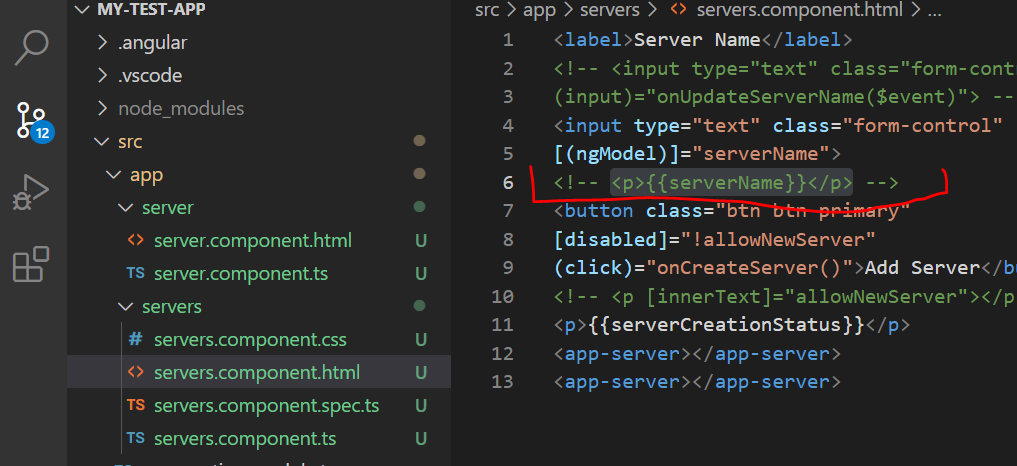
* Now if we go to the browser, we can see that the input is prepopulated with TestServer



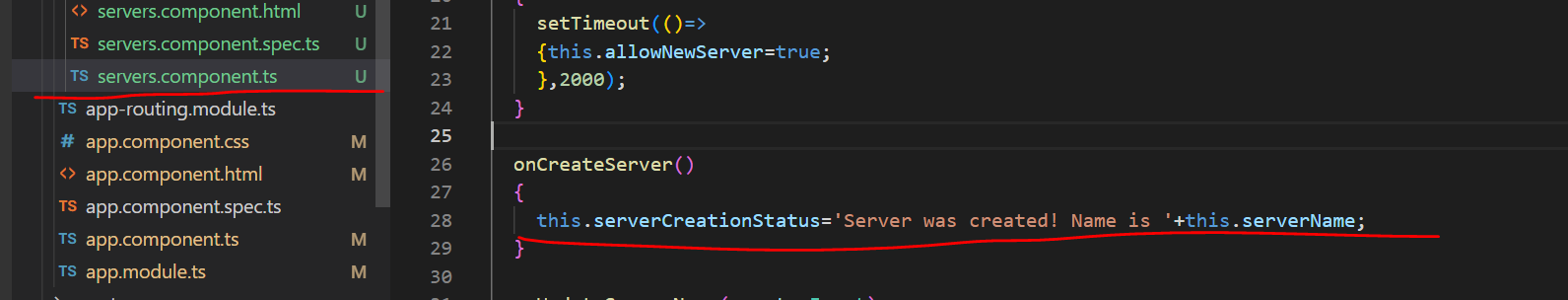
1. **Combining all forms of Databinding:**

**Requirement:**

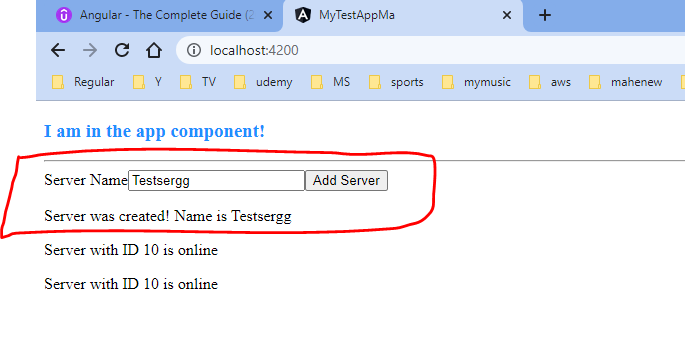
* We got the server name printed. So we shall comment the server name here



* Instead I want to click the server name, once I click the “Add Server” button.
* So once we create the server, we should not say that just “Server was created”, instead we should mention name is and add the server name



* Now with this in place, we are using the data we are changing through two-way databinding



**Four forms of databinding combined:**

* So thus far we are using all four forms of databinding:

1. With Java **Event binding** to listen to the click
2. **String Interpolation** to output the data here
3. **Property binding** to enable this button after two seconds
4. **Two way binding** to fetch our input data.
5. **Disabling a button, if username is empty:**

* We can use disabled property for the same.
* We can set the value of disabled property or use expression to set its value to true or false.
* Here we use expression to set the value of disabled.

**Click Listener to reset the user name once the “Reset User ”button is clicked:**

* On click is done , when user name is made empty and button gets disabled again

*<label>Username</label>*

*<input type="text" class="form-control" [(ngModel)]="username">*

*<p>{{ username }}</p>*

*<button*

*class="btn btn-primary"*

*[disabled]="username === ''"*

*(click)="username = ''">Reset User</button>*

*</div>*

*</div>*

*</div>*