



ANGULAR 2 FUNDAMENTALS

December 13, 2017

PART 3

DATA BINDING

DATA BINDING

- So far what we have done are only static content
- Communication between your typescript code & template (html)
- Fetch data from a server, do some calculation then display
- To output data:
 - String interpolation `{{ data }}`
 - Property binding `[property]="data"`
- React to (User) Events
 - Event binding `(event)="expression"`
 - Two way data binding `[(ngModel)] = "data"`

STRING INTERPOLATION

.....

```
<> app.component.html TS servers.component.ts x <> servers.component.html
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4    selector: 'app-servers',
5    templateUrl: './servers.component.html',
6    styleUrls: ['./servers.component.css']
7  })
8  export class ServersComponent implements OnInit {
9
10     name: string = "James";
11
12     constructor() { }
13
14     ngOnInit() {
15     }
16
17
18
19 }
```

```
<div class="container">
  <p class="sample">
    {{ name }}
  </p>
</div>
```


PROPERTY BINDING

<> servers.component.html x

```
1 | <div class="container">
2 |   <p class="sample">
3 |     {{ name }} - {{ serverId }}
4 |   </p>
5 | </div>
6 | <div class="container">
7 |   <button class="btn btn-primary" [disabled]="!allowNewServer">Add Server</button>
8 |   <p [innerText]="allowNewServer"></p>
9 | </div>
10
```

```
name: string = "James";
serverId = 10;
allowNewServer = false;

constructor() {
  setTimeout(() => {
    this.allowNewServer = true;
  }, 2000)
}

ngOnInit() {
}
```

EVENT BINDING

```
<div class="container">
  <p class="sample">
    {{ name }} - {{ serverId }}
  </p>
</div>

<div class="container">
  <button class="btn btn-primary" [disabled]="!allowNewServer" (click)="onCreateServer()">Add Server</button>
  <!-- <p [innerText]="allowNewServer"></p> -->
  <p>{{ serverCreationStatus }}</p>
</div>
```

EVENT BINDING

```
import { Component, OnInit } from '@angular/core';
import { setTimeout } from 'timers';

@Component({
  selector: 'app-servers',
  templateUrl: './servers.component.html',
  styleUrls: ['./servers.component.css']
})
export class ServersComponent implements OnInit {

  name: string = "James";
  serverId = 10;
  allowNewServer = false;
  serverCreationStatus = "No server was created!";

  constructor() {
    setTimeout(() => {
      this.allowNewServer = true;
    }, 2000)
  }

  ngOnInit() {

  }

  onCreateServer() {
    this.serverCreationStatus = "Server was created!";
  }

}
```


PASSING AND USING DATA WITH EVENT BINDING

```
@Component({
  selector: 'app-servers',
  templateUrl: './servers.component.html',
  styleUrls: ['./servers.component.css']
})
export class ServersComponent implements OnInit {

  name: string = "James";
  serverId = 10;
  allowNewServer = false;
  serverCreationStatus = "No server was created!";
  serverName = '';

  constructor() {
    setTimeout(() => {
      this.allowNewServer = true;
    }, 2000)
  }

  ngOnInit() {

  }

  onCreateServer() {
    this.serverCreationStatus = "Server Created!";
  }

  onUpdateServerName(event: any) {
    console.log(event);
    this.serverName = event.target.value;
  }
}
```

MORE ON PASSING

```
<div class="container">
<label for="">{{ serverName }}</label>
<input type="text" class="form-control" (input)="onUpdateServerName($event)">
<p class="sample">
  {{ name }} - {{ serverId }}
</p>
</div>
<div class="container">
  <button class="btn btn-primary" [disabled]="!allowNewServer" (click)="onCreateServer()">Add Server</button>
  <!-- <p [innerText]="allowNewServer"></p> -->
  <p>{{ serverCreationStatus }}</p>
</div>
```

EXERCISES

- Create an angular app
- That shows string Interpolation
- That shows property binding
- That shows event binding using click
- Use bootstrap design
- Add a input field that updates a property using two way data binding using ngModel from previous example

DIRECTIVES

- Instructions in the Dom
- Components are sample of this

NGIF TO OUTPUT DATA

- Directive based on booleans or expression

```
<div class="container">
  <label for="">{{ serverName }}</label>
  <input type="text" class="form-control" (input)="onUpdateServerName($event)">
  <p class="sample">
    | {{ name }} - {{ serverId }}
  </p>
</div>

<div class="container">
  <button class="btn btn-primary" [disabled]="!allowNewServer" (click)="onCreateServer()">Add Server</button>
  <!-- <p [innerText]="allowNewServer"></p> -->
  <p>{{ serverCreationStatus }}</p>
  <p *ngIf="serverCreated">Server was created, server name is {{ serverName }}</p>
</div>
```

MORE ON NGIF

```
export class ServersComponent implements OnInit {

  name: string = "James";
  serverId = 10;
  allowNewServer = false;
  serverCreationStatus = "No server was created!";
  serverName = '';
  serverCreated = false;

  constructor() {
    setTimeout(() => {
      this.allowNewServer = true;
    }, 2000)
  }

  ngOnInit() {

  }

  onCreateServer() {
    this.serverCreated = true;
    this.serverCreationStatus = "Server Created!";
  }
}
```


NGIF ELSE

```
<h3>If Else</h3>
```

```
<div *ngIf="isValid; else isValidFalse">
```

```
  True
```

```
</div>
```

```
<ng-template #isValidFalse>
```

```
  False
```

```
</ng-template>
```

NGIF ELSE

```
isValid = false;

constructor() {
  setTimeout(() => {
    this.allowNewServer = true;
  }, 2000)
}

ngOnInit() {
}

onCreateServer() {
  this.serverCreated = true;
  this.isValid = true;
  this.serverCreationStatus = "Server Created!";
}
```

OTHER DIRECTIVES

- `ngStyle`
- `ngClass`
- `ngFor`

END OF PART 3