



3416-003 Web Development Using Angular 8 and TypeScript

Module 6: Forms



Module 1: Section 1

What Is This Course About

Custom Property and Event Binding Summary

- Download C:\workspace-UT-Angular8-SEPT-2020\Session10\Example1-Form-Version1\bootstrap4 from google drive.
- Copy C:\workspace-UT-Angular8-SEPT-2020\Session10\Example1-Form-Version1\bootstrap4 to C:\workspace-UT-Angular8-SEPT-2020 \Session10\Example1-Form-Version2
- Examble1-Form-Version2 is based on template driven approach.

FormsModule

EXPLORER

OPEN EDITORS

- × TS *app.module.ts* src\app

BOOTSTRAP4

- > dist
- > e2e
- > node_modules
- src
 - app
 - TS *app-routing.module.ts*
 - # *app.component.css*
 - <> *app.component.html*
 - TS *app.component.spec.ts*
 - TS *app.component.ts*
 - TS *app.module.ts*

TS *app.module.ts* ×





```
src > app > TS app.module.ts
1  import { BrowserModule } from '@angular/platform-browser';
2  import { NgModule } from '@angular/core';
3
4  import { AppRoutingModule } from './app-routing.module';
5  import { AppComponent } from './app.component';
6  import { FormsModule } from '@angular/forms';
7
8  @NgModule({
9    declarations: [
10     AppComponent
11   ],
12   imports: [
13     BrowserModule,
14     AppRoutingModule,
15     FormsModule
16   ],
17   providers: [],
18   bootstrap: [AppComponent]
19 })
20 export class AppModule { }
21
```

FormsModule added

App.component.html

- Add “ngModel” and name attribute.

<> app.component.html X

vw >  div.col-xs-12.col-sm-10.col-md-8.col-sm-offset-1.col-md-offset-2 >  form >  div.form-group > 

```
1   <div class="container">
2     <div class="row">
3       <div class="col-xs-12 col-sm-10 col-md-8 col-sm-offset-1 col-md-offset-2">
4         <form>
5           <div id="user-data">
6             <div class="form-group">
7               <label for="username">Username</label>
8               <input type="text"
9                 id="username"
10                class="form-control"
11                ngModel
12                name="username">
13             </div>
14             <button class="btn btn-default" type="button">Suggest an Username</button>
15             <div class="form-group">
16               <label for="email">Mail</label>
17               <input type="email" id="email" class="form-control"
18                 ngModel
19                 name="email">
20             </div>
21           </div>
22           <div class="form-group">
23             <label for="secret">Secret Questions</label>
24             <select id="secret" class="form-control"
25               ngModel
26               name="secret">
27               <option value="pet">Your first Pet?</option>
28               <option value="teacher">Your first teacher?</option>
29             </select>
30           </div>
31           <button class="btn btn-primary" type="submit">Submit</button>
32         </form>
```

Add "ngModel" and name="username"

Add "ngModel" and name="email"

Add "ngModel" and name="secret"

Event ngSubmit

add (ngSubmit)="onSubmit(f)" #f="ngForm"



```
<> app.component.html X TS app.component.ts
src > app > <> app.component.html > div.container > div.row > div.col-xs-12.col-sm-10.col-md-8.col-sm-offset-1.col-md-offset-2 >
1 <div class="container">
2 <div class="row">
3 <div class="col-xs-12 col-sm-10 col-md-8 col-sm-offset-1 col-md-offset-2">
4 <form (ngSubmit)="onSubmit(f)" #f="ngForm">
5 <div id="user-data">
6 <div class="form-group">
7 <label for="username">Username</label>
8 <input type="text"
9 id="username"
10 class="form-control"
11 ngModel
12 name="username">
13 </div>
```

App.component.ts

import NgForm

```
<> app.component.html  TS app.component.ts X
src > app > TS app.component.ts > ...
1  import { Component } from '@angular/core';
2  import { NgForm } from '@angular/forms';
3
4  @Component({
5    selector: 'app-root',
6    templateUrl: './app.component.html',
7    styleUrls: ['./app.component.css']
8  })
9  export class AppComponent {
10    suggestUserName() {
11      | const suggestedName = 'Superuser';
12    }
13
14    onSubmit(form: NgForm) {
15      | console.log(form);
16    }
17  }
18
```

add this method onSubmit

Tested it on browser

[INFO] LIVE RELOADING ENABLED.

```
▼ NgForm {submitted: true, _directives: Array(3), ngSubmit: EventEmitter, form: FormGroup}
  formDirective: (...)
    control: (...)
    path: (...)
    controls: (...)
    value: (...)
    valid: (...)
    invalid: (...)
    pending: (...)
    disabled: (...)
    enabled: (...)
    errors: (...)
    pristine: (...)
    dirty: (...)
    touched: (...)
    status: (...)
    untouched: (...)
    statusChanges: (...)
    valueChanges: (...)
    submitted: true
  _directives: (3) [NgModel, NgModel, NgModel]
  ngSubmit: EventEmitter {_isScalar: false, observers: Array(1), closed: false, isStopped
  form: FormGroup
    parent: (...)
    valid: (...)
    invalid: (...)
    pending: (...)
    disabled: (...)
    enabled: (...)
    dirty: (...)
    untouched: (...)
    updateOn: (...)
    root: (...)
    validator: null
    asyncValidator: null
    _onCollectionChange: () => { }
    pristine: false
    touched: true
    _onDisabledChange: []
    controls: {username: FormControl, email: FormControl, secret: FormControl}
    valueChanges: EventEmitter {_isScalar: false, observers: Array(0), closed: false, isS
    statusChanges: EventEmitter {_isScalar: false, observers: Array(0), closed: false, is
    status: "VALID"
    value:
      username: "jagade100"
      email: "jagade100@gmail.com"
      secret: "pet"
      __proto__: Object
    errors: null
    __proto__: AbstractControl
    __proto__: ControlContainer
```

Click "form".

Click "value" and get username, email and secret

Understanding Form State

- Copy C:\workspace-UT-Angular8-SEPT-2020
\Session10\Example1-Form-Version2\bootstrap4 to C:\
workspace-UT-Angular8-SEPT-2020\Session10\Example1-
Form-Version3

Properties of control

```
▶ _directives: (3) [NgModel, NgModel, NgModel]
▶ ngSubmit: EventEmitter {_isScalar: false, observers: Array(1), closed: false, isStopped: false, hasError: false, ...}
▼ form: FormGroup
  parent: (...)
  valid: (...)
  invalid: (...)
  pending: (...)
  disabled: (...)
  enabled: (...)
  dirty: (...)
  untouched: (...)
  updateOn: (...)
  root: (...)
  validator: null
  asyncValidator: null
▶ _onCollectionChange: () => { }
pristine: false
touched: true
▶ _onDisabledChange: []
▼ controls:
  ▼ username: FormControl
    parent: (...)
    valid: true
    invalid: false
    pending: (...)
    disabled: false
    enabled: (...)
    dirty: true
    untouched: false
    updateOn: (...)
    root: (...)
    validator: null
    asyncValidator: null
    ▶ _onCollectionChange: () => { }
    pristine: false
    touched: true
    ▶ _onDisabledChange: [f]
    ▶ _onChange: [f]
    _pendingValue: "jadeite100"
    value: "jadeite100"
    status: "VALID"
    errors: null
    ▶ valueChanges: EventEmitter {_isScalar: false, observers: Array(0), closed: false, isStopped: false, hasError: false, ...}
    ▶ statusChanges: EventEmitter {_isScalar: false, observers: Array(0), closed: false, isStopped: false, hasError: false, ...}
    ▶ _parent: FormGroup {validator: null, asyncValidator: null, pristine: false, touched: true, _onCollectionChange: f, ...}
    _pendingChange: false
    _pendingDirty: true
    _pendingTouched: true
    ▶ __proto__: AbstractControl
    ▶ email: FormControl {validator: null, asyncValidator: null, pristine: false, touched: true, _onCollectionChange: f, ...}
    ▶ secret: FormControl {validator: null, asyncValidator: null, pristine: false, touched: true, _onCollectionChange: f, ...}
    ▶ __proto__: Object
  ▶ valueChanges: EventEmitter { _isScalar: false, observers: Array(0), closed: false, isStopped: false, hasError: false, ... }
```

Value changed based on user input



Properties of control

- Properties of control
 - username
 - Dirty ->true ->This property returns true if the element's contents have been changed. Example if the user typed something in the username field.
 - untouched->false->This property returns true if the user has not visited OR touched the element.
 - Invalid->false->The user entered something into the field.
 - Valid->true->it is a valid field because the user entered a value.

Add attributes for directives

```
<div class="row">
  <div class="col-xs-12 col-sm-10 col-md-8 col-sm-offset-1 col-md-offset-2">
    <form (ngSubmit)="onSubmit(f)" #f="ngForm">
      <div id="user-data">
        <div class="form-group">
          <label for="username">Username</label>
          <input type="text"
            id="username"
            class="form-control"
            ngModel
            name="username"
            required>
          add "required" attribute
        </div>
        <button class="btn btn-default" type="button">Suggest an Username</button>
        <div class="form-group">
          <label for="email">Mail</label>
          <input type="email" id="email" class="form-control"
            ngModel
            name="email"
            required
            email>
          add "required" and "email" attribute.
        </div>
      </div>
    </div>
    <div class="form-group">
      <label for="secret">Secret Questions</label>
      <select id="secret" class="form-control"
        ngModel
        name="secret">
        <option value="pet">Your first Pet?</option>
        <option value="teacher">Your first teacher?</option>
      </select>
      add this code
    </div>
    <button class="btn btn-primary"
      type="submit"
      [disabled]="!f.valid">Submit</button>
```

app.component.css

- Edit C:\workspace-UT-Angular8-SEPT-2020
 \Session10\Example1-Form-
 Version3\bootstrap4\src\app\app.component.css

```
.container {  
  margin-top: 30px;  
}
```

```
input.ng-invalid.ng-touched {  
  border: 1px solid red;  
}
```

app.component.css

- Edit C:\workspace-UT-Angular8-SEPT-2020
 \Session10\Example1-Form-
 Version3\bootstrap4\src\app\app.component.css

<> app.component.html

app.component.css X

src > app > # app.component.css >  input.ng-invalid.ng-touched

```
1  .container {  
2  |   margin-top: 30px;  
3  | }  
4  
5  input.ng-invalid.ng-touched {  
6  |   border: 1px solid  red;  
7  | }  
8
```

app.component.html

- Copy C:\workspace-UT-Angular8-SEPT-2020 \Session10\Example1-Form-Version3\bootstrap4 to C:\workspace-UT-Angular8-SEPT-2020 \Session10\Example1-Form-Version4
- Edit app.component.html:

```
<div class="form-group">  
  <label for="email">Mail</label>  
  <input type="email" id="email" class="form-control"  
    ngModel  
    name="email"  
    required  
    email  
    #email="ngModel">  
    <span class="help-block" *ngIf="email.valid &&  
email.touched">Please enter a valid email</span>  
</div>
```

app.component.html

- Add a span tag for the email field.

```
<div class="col-xs-12 col-sm-10 col-md-8 col-sm-offset-1 col-md-offset-2">
  <form (ngSubmit)="onSubmit(f)" #f="ngForm">
    <div id="user-data">
      <div class="form-group">
        <label for="username">Username</label>
        <input type="text"
          id="username"
          class="form-control"
          ngModel
          name="username"
          required>
      </div>
      <button class="btn btn-default" type="button">Suggest an Username</button>
      <div class="form-group">
        <label for="email">Mail</label>
        <input type="email" id="email" class="form-control"
          ngModel
          name="email"
          required
          email
          #email="ngModel">
        <span class="help-block" *ngIf="email.invalid && email.touched">Please enter a valid
      </div>
    </div>
    <div class="form-group">
      <label for="secret">Secret Questions</label>
      <select id="secret" class="form-control"
        ngModel
        name="secret">
        <option value="pet">Your first Pet?</option>
        <option value="teacher">Your first teacher?</option>
      </select>
    </div>
  </form>
</div>
```

Entered this line

Set Default Values with ngModel Property Bindig.

- Copy C:\workspace-UT-Angular8-SEPT-2020
\Session10\Example1-Form-Version4\bootstrap4 to C:\
workspace-UT-Angular8-SEPT-2020\Session10\Example1-
Form-Version5

app.component.ts

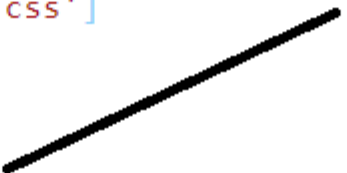
- Edit app.component.ts

```
TS app.component.ts X  <> app.component.html  TS app.c

src > app > TS app.component.ts > AppComponent > defaultQuestion

1  import { Component } from '@angular/core';
2  import { NgForm } from '@angular/forms';
3
4  @Component({
5      selector: 'app-root',
6      templateUrl: './app.component.html',
7      styleUrls: ['./app.component.css']
8  })
9  export class AppComponent {
10
11      defaultQuestion = 'teacher';
12
13      suggestUserName() {
14          const suggestedName = 'Superuser';
15      }
16
17      onSubmit(form: NgForm) {
18          console.log(form);
19      }
20  }
```

Add this line



Edit app.component.html file

```
<div class="container">
  <div class="row">
    <div class="col-xs-12 col-sm-10 col-md-8 col-sm-offset-1 col-md-offset-2">
      <form (ngSubmit)="onSubmit(f)" #f="ngForm">
        <div id="user-data">
          <div class="form-group">
            <label for="username">Username</label>
            <input type="text"
              id="username"
              class="form-control"
              ngModel
              name="username"
              required>
          </div>
          <button class="btn btn-default" type="button">Suggest an Username</button>
          <div class="form-group">
            <label for="email">Mail</label>
            <input type="email" id="email" class="form-control"
              ngModel
              name="email"
              required
              email
              #email="ngModel">
            <span class="help-block" *ngIf="email.valid && email.touched">Please enter a valid e
          </div>
        </div>
        <div class="form-group">
          <label for="secret">Secret Questions</label>
          <select id="secret"
            class="form-control"
            [ngModel]="defaultQuestion"
            name="secret">
            <option value="pet">Your first Pet?</option>
            <option value="teacher">Your first teacher?</option>
          </select>
        </div>
      </form>
    </div>
  </div>
</div>
```

change to "[ngModel] property
and add "defaultQuestion"

Edit app.component.ts file

```
TS app.component.ts X app.component.html
src > app > TS app.component.ts > AppComponent >
1  import { Component } from '@angular/core';
2  import { NgForm } from '@angular/forms';
3
4  @Component({
5    selector: 'app-root',
6    templateUrl: './app.component.html',
7    styleUrls: ['./app.component.css']
8  })
9  export class AppComponent {
10
11    defaultQuestion = 'teacher';
12
13    suggestUserName() {
14      | const suggestedName = 'Superuser';
15    }
16
17    onSubmit(form: NgForm) {
18      | console.log(form);
19    }
20
21
```

Add the "defaultQuestion"
variable

Using ngModel with Two-Way-Binding

- Copy C:\workspace-UT-Angular8-SEPT-2020
 \Session5\Example1-Form-Version5\bootstrap4 to
 C:\workspace-UT-Angular8-SEPT-
 2020\Session10\Example1-Form-Version6\

Using ngModel with Two-Way-Binding

- Edit app.component.s

```
<> app.component.html ● TS app.component.ts ✕

src > app > TS app.component.ts > 📁 AppComponent
1  import { Component } from '@angular/core';
2  import { NgForm } from '@angular/forms';
3
4  @Component({
5    selector: 'app-root',
6    templateUrl: './app.component.html',
7    styleUrls: ['./app.component.css']
8  })
9  export class AppComponent {
10
11    defaultQuestion = 'teacher';
12    answer = '';
13
14    suggestUserName() {
15      | const suggestedName = 'Superuser';
16    }
17
18    onSubmit(form: NgForm) {
19      | console.log(form);
20    }
21  }
```

add the variable answer.

Using ngModel with Two-Way-Binding

- Edit app.component.ts:

```
<> app.component.html ● TS app.component.ts
src > app > <> app.component.html > div.container > div.row > div.col-xs-6
21         required
22         email
23         #email="ngModel">
24         <span class="help-block" *ngIf="email.valid && email.touched">Please e
25     </div>
26 </div>
27 <div class="form-group">
28     <label for="secret">Secret Questions</label>
29     <select id="secret"
30         class="form-control"
31         [ngModel]="defaultQuestion"
32         name="secret">
33         <option value="pet">Your first Pet?</option>
34         <option value="teacher">Your first teacher?</option>
35     </select>
36 </div>
37 <div class="form-group">
38     <textarea
39         name="questionAnswer"
40         row="3"
41         class="form-control"
42         [(ngModel)]="answer"></textarea>
43 </div>
44 <p>Your reply: {{ answer }} </p>
45 <button class="btn btn-primary"
46     type="submit"
47     [disabled]="!f.valid">Submit</button>
48 </form>
49 </div>
50 </div>
51 </div>
52
```

Add these new lines of code

Using ngModel with Two-Way-Binding

Bootstrap4 x +

localhost:4200

Apps Sunnybrook Sunnybrook JackNathan typescript HouseForSale YouTubeConverter china Imported Jenkins Timesheet GoogleDrive epub-to-pdf what-is-my-ip

Username

Suggest an Username

Mail

Please enter a valid email

Secret Questions

Your first Pet? ▼

Your reply: Neilly

Submit

Grouping Form Controls

- Copy C:\workspace-UT-Angular8-SEPT-2020
\Session5\Example1-Form-Version6\bootstrap4 to
C:\workspace-UT-Angular8-SEPT-2020
\Session10\Example1-Form-Version7\

```
<> app.component.html X Settings
src > app > <> app.component.html > div.container > div.row > div.col-xs-12.col-sm-10.col-md-8.col-sm-offset-1.col-md-offset-2
1  <div class="container">
2    <div class="row">
3      <div class="col-xs-12 col-sm-10 col-md-8 col-sm-offset-1 col-md-offset-2">
4        <form (ngSubmit)="onSubmit(f)" #f="ngForm">
5          <div id="user-data" ngModelGroup="userData">
6            <div class="form-group">
7              <label for="username">Username</label>
8              <input type="text"
9                id="username"
10               class="form-control"
11               ngModel
12               name="username"
13               required>
14            </div>
15            <button class="btn btn-default" type="button">Suggest an Username</button>
16            <div class="form-group">
17              <label for="email">Mail</label>
18              <input type="email" id="email" class="form-control"
19                ngModel
20                name="email"
21                required
22                email
23                #email="ngModel">
24              <span class="help-block" *ngIf="email.valid && email.touched">Please enter a valid email</span>
25            </div>
26          </div>
27          <div class="form-group">
28            <label for="secret">Secret Questions</label>
29            <select id="secret"
30              class="form-control"
31              [ngModel]="defaultQuestion"
```

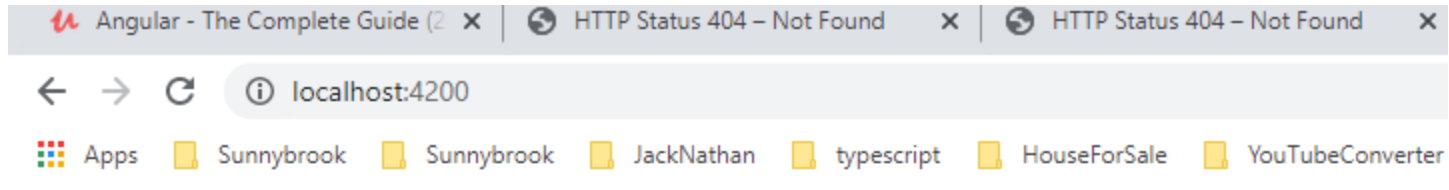
Grouping Form Controls

- Edit application.component.html:

```
<> app.component.html X Settings
component.html > div.container > div.row > div.col-xs-12.col-sm-10.col-md-8.col-sm-offset-1.col-md-offset-2 > for
1  <div class="container">
2    <div class="row">
3      <div class="col-xs-12 col-sm-10 col-md-8 col-sm-offset-1 col-md-offset-2">
4        <form (ngSubmit)="onSubmit(f)" #f="ngForm">
5          <div id="user-data" ngModelGroup="userData">
6            <div class="form-group">
7              <label for="username">Username</label>
8              <input type="text"
9                id="username"
10               class="form-control"
11               ngModel
12               name="username"
13               required>
14            </div>
15            <button class="btn btn-default" type="button">Suggest an Username</button>
16            <div class="form-group">
17              <label for="email">Mail</label>
18              <input type="email" id="email" class="form-control"
19                ngModel
20                name="email"
21                required
22                email
23                #email="ngModel">
24              <span class="help-block" *ngIf="email.valid && email.touched">Please enter a valid email</span>
25            </div>
26          </div>
27          <div class="form-group">
28            <label for="secret">Secret Questions</label>
29            <select id="secret"
30              class="form-control"
31              [ngModel]="defaultQuestion">
```

Grouping Form Controls

- google chrome:



Username

Suggest an Username

Mail

Please enter a valid email

Secret Questions

Your first Pet?

Einstein

Grouping Form Controls

- Console in google chrome:

```
app.component.ts:
▼ NgForm ⓘ
  formDirective: (...)
  control: (...)
  path: (...)
  controls: (...)
  value: (...)
  valid: (...)
  invalid: (...)
  pending: (...)
  disabled: (...)
  enabled: (...)
  errors: (...)
  pristine: (...)
  dirty: (...)
  touched: (...)
  status: (...)
  untouched: (...)
  statusChanges: (...)
  valueChanges: (...)
  submitted: true
  _directives: (4) [NgModel, NgModel, NgModel, NgModel]
  ngSubmit: EventEmitter {_isScalar: false, observers: Array(1), closed: false, isStopped: false, hasError: false, ...}
  ▼ form: FormGroup
    parent: (...)
    valid: (...)
    invalid: (...)
    pending: (...)
    disabled: (...)
    enabled: (...)
    dirty: (...)
    untouched: (...)
    updateOn: (...)
    root: (...)
    validator: null
    asyncValidator: null
    _onCollectionChange: () => { }
    pristine: false
    touched: true
    _onDisabledChange: []
    controls: {userData: FormGroup, secret: FormControl, questionAnswer: FormControl}
```

Grouping Form Controls

- Console in google chrome(controls, userData):

```
app.component.ts:19
▼ NgForm 1
  formDirective: (...)
  ▼ control: FormGroup
    parent: (...)
    valid: (...)
    invalid: (...)
    pending: (...)
    disabled: (...)
    enabled: (...)
    dirty: (...)
    untouched: (...)
    updateOn: (...)
    root: (...)
    validator: null
    asyncValidator: null
    ▶ _onCollectionChange: () => { }
    pristine: false
    touched: true
    ▶ _onDisabledChange: [?]
    ▼ controls:
      ▶ userData: FormGroup {validator: null, asyncValidator: null, pristine: false, touched: true, _onCollectionChange: f, ...}
      ▶ secret: FormControl {validator: null, asyncValidator: null, pristine: false, touched: true, _onCollectionChange: f, ...}
      ▶ questionAnswer: FormControl {validator: null, asyncValidator: null, pristine: false, touched: true, _onCollectionChange: f, ...}
      ▶ __proto__: Object
    ▶ valueChanges: EventEmitter {_isScalar: false, observers: Array(0), closed: false, isStopped: false, hasError: false, ...}
    ▶ statusChanges: EventEmitter {_isScalar: false, observers: Array(0), closed: false, isStopped: false, hasError: false, ...}
    status: "VALID"
    ▶ value: {userData: {...}, secret: "pet", questionAnswer: "Einstein"}
    errors: null
    ▶ __proto__: AbstractControl
  path: (...)
  controls: (...)
  value: (...)
```

controls "userData"

Grouping Form Controls

- Elements in google chrome(controls, userData):

Username

Suggest an Username

Mail

Please enter a valid email

Secret Questions

Your first Pet? ▼

Einstein

Your reply: Einstein

Submit

```
<!doctype html>
<html lang="en">
  <head>...</head>
  <body>
    <app-root _ngcontent-vxa-c0 ng-version="8.2.14">
      <div _ngcontent-vxa-c0 class="container">
        <div _ngcontent-vxa-c0 class="row">
          <div _ngcontent-vxa-c0 class="col-xs-12 col-sm-10 col-md-8 col-sm-offset-1 col-md-offset-2">
            <form _ngcontent-vxa-c0 novalidate class="ng-dirty ng-touched ng-valid">
              <div _ngcontent-vxa-c0 id="user-data" ngmodelgroup="userData" ng-reflect-name="userData" class="ng-dirty ng-touched ng-valid"> == $0
                <div _ngcontent-vxa-c0 class="form-group">
                  <label _ngcontent-vxa-c0 for="username">Username</label>
                  <input _ngcontent-vxa-c0 class="form-control ng-dirty ng-valid ng-touched" id="username" name="username" ngmodel required type="text" ng-reflect-required ng-reflect-name="username" ng-reflect-model>
                </div>
                <button _ngcontent-vxa-c0 class="btn btn-default" type="button">Suggest an Username</button>
              <div _ngcontent-vxa-c0 class="form-group">...</div>
            </div>
            <div _ngcontent-vxa-c0 class="form-group">...</div>
            <div _ngcontent-vxa-c0 class="form-group">...</div>
            <p _ngcontent-vxa-c0>Your reply: Einstein </p>
            <button _ngcontent-vxa-c0 class="btn btn-primary" type="submit">Submit</button>
          </form>
        </div>
      </div>
    </app-root>
    <script src="runtime.js" type="module"></script>
    <script src="polyfills.js" type="module"></script>
```

Grouping Form Controls(How to use userData in javascript in the html page)

- Copy C:\workspace-UT-Angular8-SEPT-2020 \Session5\Example1-Form-Version7\bootstrap4 to C:\workspace-UT-Angular8-SEPT-2020 \Session10\Example1-Form-Version8\
- How to use “userData” in the javascript in the html.



```
> app.component.html X
rc > app > <> app.component.html > div.container > div.row > div.col-xs-12.col-sm-10.col-md-8.col-sm-offset-1.co
1 <div class="container">
2   <div class="row">
3     <div class="col-xs-12 col-sm-10 col-md-8 col-sm-offset-1 col-md-offset-2">
4       <form (ngSubmit)="onSubmit(f)" #f="ngForm">
5         <div
6           id="user-data"
7           ngModelGroup="userData"
8           #userData="ngModelGroup">
9         <div class="form-group">
10          <label for="username">Username</label>
11          <input type="text"
12            id="username"
13            class="form-control"
14            ngModel
15            name="username"
16            required>
17        </div>
```

Grouping Form Controls(How to use userData in javascript in the html page)

- Copy C:\workspace-UT-Angular8-SEPT-2020 \Session5\Example1-Form-Version7\bootstrap4 to C:\workspace-UT-Angular8-SEPT-2020 \Session10\Example1-Form-Version8\
- How to use “userData” in the javascript in the html.



```
> app.component.html X
rc > app > <> app.component.html > div.container > div.row > div.col-xs-12.col-sm-10.col-md-8.col-sm-offset-1.co
1 <div class="container">
2   <div class="row">
3     <div class="col-xs-12 col-sm-10 col-md-8 col-sm-offset-1 col-md-offset-2">
4       <form (ngSubmit)="onSubmit(f)" #f="ngForm">
5         <div
6           id="user-data"
7           ngModelGroup="userData"
8           #userData="ngModelGroup">
9           <div class="form-group">
10             <label for="username">Username</label>
11             <input type="text"
12               id="username"
13               class="form-control"
14               ngModel
15               name="username"
16               required>
17           </div>
```


Grouping Form Controls(How to use userData in javascript in the html page)

- How to use “userData” in the javascript in the html.

<> app.component.html X

t.html > div.container > div.row > div.col-xs-12.col-sm-10.col-md-8.col-sm-offset-1.col-md-offset-2 > form

```
1 <div class="container">
2   <div class="row">
3     <div class="col-xs-12 col-sm-10 col-md-8 col-sm-offset-1 col-md-offset-2">
4       <form (ngSubmit)="onSubmit(f)" #f="ngForm">
5         <div
6           id="user-data"
7           ngModelGroup="userData"
8           #userData="ngModelGroup">
9           <div class="form-group">
10            <label for="username">Username</label>
11            <input type="text"
12              id="username"
13              class="form-control"
14              ngModel
15              name="username"
16              required>
17          </div>
18          <button class="btn btn-default" type="button">Suggest an Username</button>
19          <div class="form-group">
20            <label for="email">Mail</label>
21            <input type="email" id="email" class="form-control"
22              ngModel
23              name="email"
24              required
25              email
26              #email="ngModel">
27            <span class="help-block" *ngIf="email.valid && email.touched">Please enter a valid email</span>
28          </div>
29        </div>
30        <p *ngIf="!userData.valid && userData.touched">User Data is invalid!</p>
31        <div class="form-group">
32          <label for="secret">Secret Questions</label>
33          <select id="secret">
```

add this line

Grouping Form Controls(How to use userData in javascript in the html page)

- Click on “Username” field and leave it:

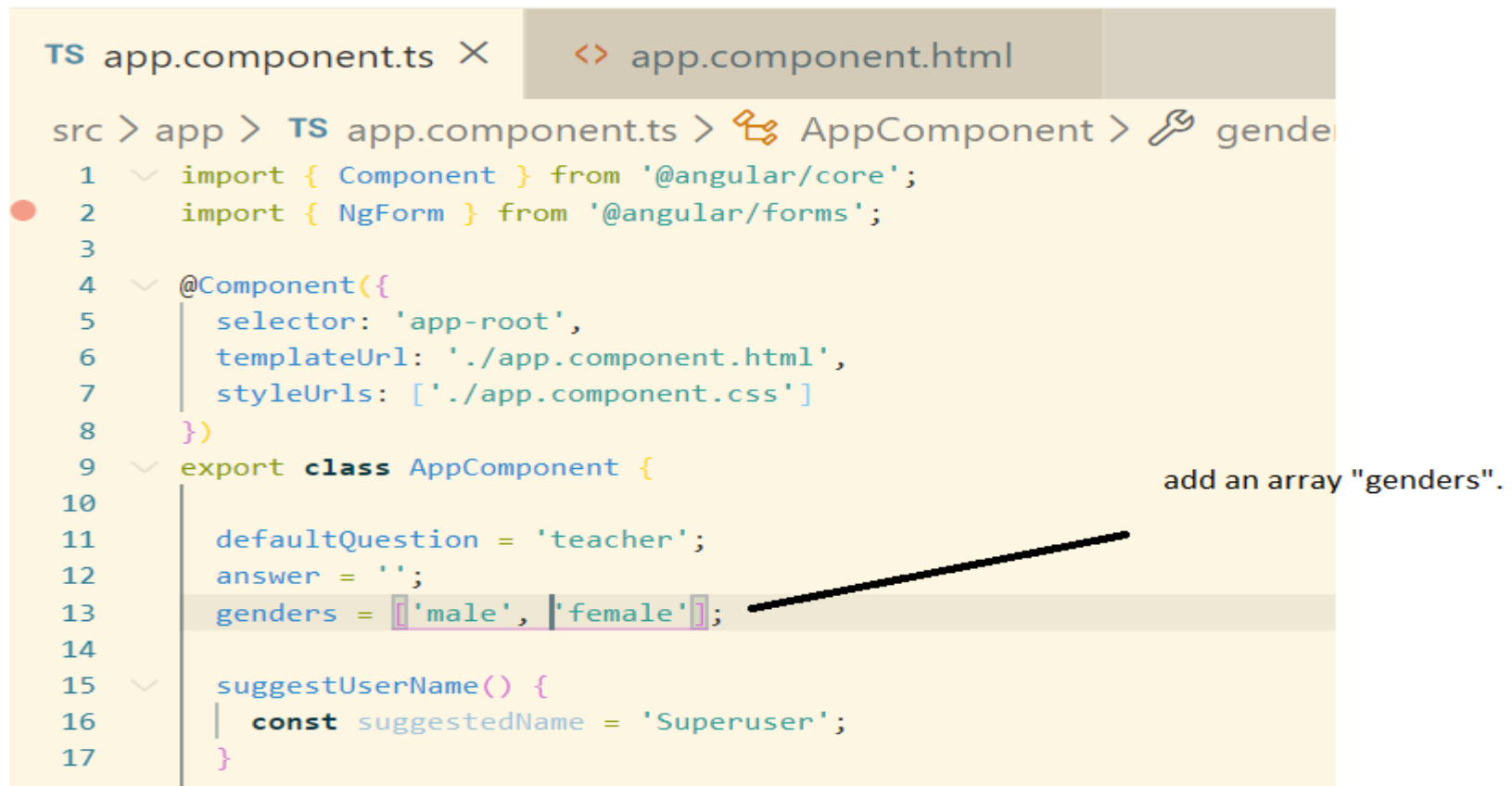
The screenshot shows a web browser window with the address bar displaying 'localhost:4200'. The browser tabs include 'Angular - The Complete Guide (2 x)', 'HTTP Status 404 - Not Found', and 'Int'. The browser's bookmark bar shows 'Apps', 'Sunnybrook', 'Sunnybrook', 'JackNathan', 'typescript', 'HouseForSale', 'YouTubeConverter', and 'chi'. The form contains the following elements:

- Username**: A text input field with a red border.
- Suggest an Username**: A button below the Username field.
- Mail**: A text input field.
- User Data is invalid!**: A message displayed below the Mail field.
- Secret Questions**: A section containing a dropdown menu with the text 'Your first teacher?' and a small downward arrow.
- Empty text area**: A large, empty text area at the bottom of the form.

Two black arrows point from the text 'Click on "Username" field and click outside, you will see message "User Data is invalid!"' to the Username input field and the 'User Data is invalid!' message.

Radio buttons

- Copy C:\workspace-UT-Angular8-SEPT-2020 \Session5\Example1-Form-Version8\bootstrap4 to C:\workspace-UT-Angular8-SEPT-2020 \Session10\Example1-Form-Version9\



```
TS app.component.ts X <> app.component.html
src > app > TS app.component.ts > AppComponent > genders
1  import { Component } from '@angular/core';
2  import { NgForm } from '@angular/forms';
3
4  @Component({
5    selector: 'app-root',
6    templateUrl: './app.component.html',
7    styleUrls: ['./app.component.css']
8  })
9  export class AppComponent {
10
11    defaultQuestion = 'teacher';
12    answer = '';
13    genders = ['male', 'female'];
14
15    suggestUserName() {
16      const suggestedName = 'Superuser';
17    }
18  }
```

add an array "genders".

Radio buttons

- Copy C:\workspace-UT-Angular8-SEPT-2020 \Session10\Example1-Form-Version8\bootstrap4 to C:\workspace-UT-Angular8-SEPT-2020 \Session10\Example1-Form-Version9\

```
TS app.component.ts    <> app.component.html X
```

```
container > div.row > div.col-xs-12.col-sm-10.col-md-8.col-sm-offset-1.col-md-offset-2 >
```

```
32 <label for= "secret" >Secret Questions</label>
33 <select id="secret"
34   class="form-control"
35   [ngModel]="defaultQuestion"
36   name="secret">
37   <option value="pet">Your first Pet?</option>
38   <option value="teacher">Your first teacher?</option>
39 </select>
40 </div>
41 <div class="form-group">
42   <textarea
43     name="questionAnswer"
44     row="3"
45     class="form-control"
46     [(ngModel)]="answer"></textarea>
47 </div>
48 <p>Your reply: {{ answer }} </p>
49 <div class="radio" *ngFor="let gender of genders ">
50   <label>
51     <input
52       type="radio"
53       name="gender"
54       ngModel
55       [value]="gender">
56     {{ gender }}
57   </label>
58 </div>
59 <button class="btn btn-primary"
60   type="submit"
61   [disabled]="!f.valid">Submit</button>
62 </form>
```

Add these line of code

Radio buttons

Username

Suggest an Username

Mail

Please enter a valid email

Secret Questions

Your first teacher?

Your reply:

☒ male

☐ female

Submit

Enter the values and
click "Submit" and see
gender="male" in value
object

```
NgForm {submitted: true, _directives: Array(6), ngSubmit: EventEmitter, form: FormGroup}
  formDirective: (...)
  control: (...)
  path: (...)
  controls: (...)
  value: (...)
  valid: (...)
  invalid: (...)
  pending: (...)
  disabled: (...)
  enabled: (...)
  errors: (...)
  pristine: (...)
  dirty: (...)
  touched: (...)
  status: (...)
  untouched: (...)
  statusChanges: (...)
  valueChanges: (...)
  submitted: true
  _directives: (6) [NgModel, NgModel, NgModel, NgModel, NgModel, NgModel]
  ngSubmit: EventEmitter {_isScalar: false, observers: Array(1), closed: false, isStopped: false, hasError: false, ...}
  form: FormGroup
    parent: (...)
    valid: (...)
    invalid: (...)
    pending: (...)
    disabled: (...)
    enabled: (...)
    dirty: (...)
    untouched: (...)
    updateOn: (...)
    root: (...)
    validator: null
    asyncValidator: null
    _onCollectionChange: () => { }
    pristine: false
    touched: true
    _onDisabledChange: []
    controls: {userData: FormGroup, secret: FormControl, questionAnswer: FormControl, gender: FormControl}
    valueChanges: EventEmitter {_isScalar: false, observers: Array(0), closed: false, isStopped: false, hasError: false, ...}
    statusChanges: EventEmitter {_isScalar: false, observers: Array(0), closed: false, isStopped: false, hasError: false, ...}
    status: "VALID"
    value:
      userData: {username: "Teset", email: "test@test.com"}
      secret: "teacher"
      questionAnswer: ""
      gender: "male"
      __proto__: Object
    errors: null
    __proto__: AbstractControl
    proto: ControlContainer
```