**Lesson19 Angular form array example**

**Notes:-**

**1-To build an Angular Reactive Form we use three fundamental building blocks**

**A-Form Control B-Form Group C-Form Array**

**2-Form Array as the name implies is an array. It can contain an array of**

**A-Form Controls B-Form Groups C-Nested Form Arrays**

**Form Arrays can hold arrays of items such as form control, unlike Arrays can store like Items.**

**We have a Form Array with:-**  
**1-one Form Control 2-one Form Group 3-one Form Array**

**Example:-**

**The following example will show the form Array**

onLoadDataClick(){

//we make Form Array that contain Form Control, Form Group, Form Array

const formArray = new FormArray([

new FormControl('John',Validators.required),

new FormGroup({

country:new FormControl('',Validators.required)}),

new FormArray([])])

this.loadControlsAndGroup(formArray);}

loadControlsAndGroup(lst:FormArray){

//we make for loop of controls to detect the type of control

for(const item of lst.controls){

if(item instanceof FormControl){console.log('This is form control');}

if(item instanceof FormGroup){console.log('This is form group');}

if(item instanceof FormArray){console.log('This Is form array');}}}

**Example:-**

**There are another way to define Form Array**

onLoadDataClick(){

//There is another way to define Form Array

const formArray = this.fb.array([

new FormControl('John',Validators.required),

new FormGroup({

country:new FormControl('',Validators.required)}),

new FormArray([])])

//The following command is to get the value ["John", {…}, Array(0)]

console.log(formArray.value);

this.loadControlsAndGroup(formArray);}

loadControlsAndGroup(lst:FormArray){

for(const item of lst.controls){

if(item instanceof FormControl){

console.log('This '+item.value+' is form control');}

if(item instanceof FormGroup){

console.log('This '+item.value+' is form group');}

if(item instanceof FormArray){

console.log('This '+item.value+' Is form array');}}}

**Notes:-**

**We usually use the following properties to determine the state of a Form Control or a Form Group. These status properties are also available on a Form Array. For example, if one of the controls in a Form Array is touched, the entire array becomes touched. Similarly, if one of the controls is invalid, the entire array becomes invalid.**

**1-touched 2-untouched 3-dirty**

**4-pristine 5-valid 6-invalid**

**Useful FormArray methods**

|  |  |
| --- | --- |
| **Method** | **Purpose** |
| push | Inserts the control at the end of the array |
| insert | Inserts the control at the specified index in the array |
| removeAt | Removes the control at the specified index in the array |
| setControl | Replace an existing control at the specified index in the array |
| at | Return the control at the specified index in the array |

**We can also use a FormGroup to create a group of FormControls. Notice, in the example below, we are using the group() method of the FormBuilder class to create a FormGroup.**

const formGroup = this.fb.group([

  new FormControl('John', Validators.required),

  new FormControl('IT', Validators.required),

  new FormControl('', Validators.required),]);

**What is the difference between a Form Group and a Form Array?**

**1-FormArray data is serialized as an array where as a Form Group is serialized as an object.**

console.log(formArray.value);   
**Output:** [FormControl, FormControl, FormControl]

console.log(formGroup.value);  
**Output:** {0: FormControl, 1: FormControl, 2: FormControl}

**2-Form Array have good benefit to create dynamic elements such as create skills elements dynamically**