Angular

Tahaluf Training Center 2021

Chapter 2

1. What is Data Binding
2. One way data binding
3. Two way data binding
4. How to create module in angular

Data Binding

Allows to define communication between a component and the DOM, making it very easy to define interactive applications without worrying about pushing and pulling information.

Chapter 2

1. What is Data Binding
2. One way data binding
3. Two way data binding
4. How to create module in angular

One way data binding

One-way data binding will bind the data from the component to the view (DOM) or from view to the component .

One way data binding

One way data binding may be:

Input event → Read event .

OR

Output event → Write event .

One way data binding

You can use this way to read the value from variable.

In app.componemts.html

```
<input type ="text" placeholder="your name"
value="{{name}}" />
```

OR (using bracket)

```
<input type="text"  placeholder="your name"
 [value]="name" />
```

One way data binding

You can use this way to read the value from variable

In app.component.html

```
<input type="text" placeholder="your name"
[value]="name" (change)="handleNameInputChange()" />
```

In app.component.ts

```
onChange(){
  alert("The value is changed!")
}
```

Event Object :

You can display the input value by binding key event and displays the text back what the user types onto the screen.

In app.componemts.html

```
<input type="text" placeholder="your name"
[value]="name"
(change)="handleNameInputChange($event)" />
```

In app.componemts.ts

```
handleNameInputChange = (e:any) =>
{
   console.log(e.target.value)
}
```

Chapter 3

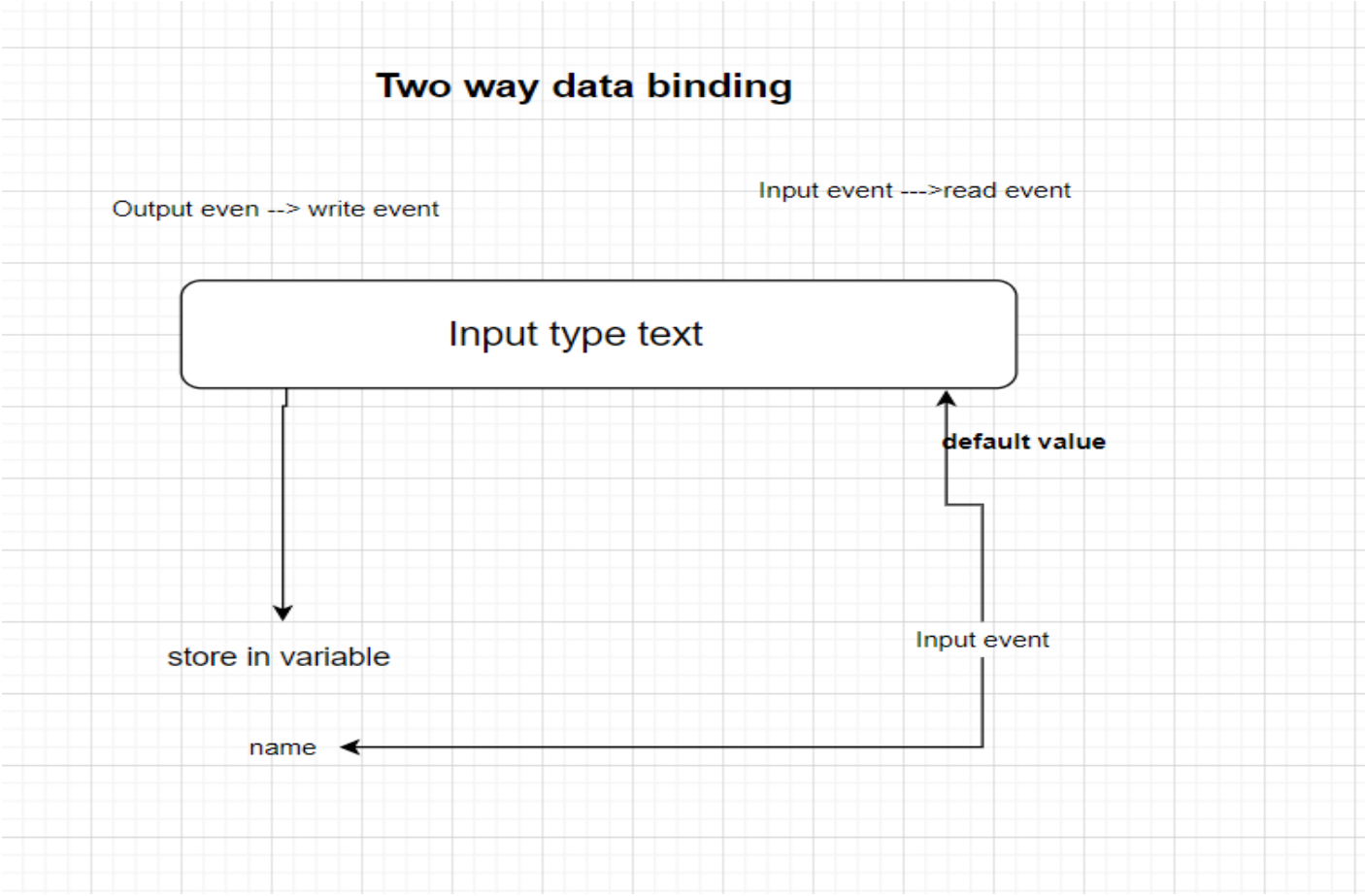
1. What is Data Binding
2. One way data binding
3. Two way data binding
4. How to create module in angular

Two-way data binding

Allows to have the data flow both ways (Read and Write event).

And it is a continuous synchronization of a data from view to the component and component to the view.

Two way data binding



Two way data binding

In two way data binding we will use Ngmodel Which creates a FormControl instance and binds it to a form control element.

First we will add the Forms model in app.module.ts in import section.

```
import { FormsModule } from '@angular/forms';

imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule
],
```

Lets have a demo

Creates a simple form using two way data binding which contains :

- ✓ Name
- ✓ Email
- ✓ Salary
- ✓ And then calculate the annual salary.

Two way data binding

To use two way data binding you must use [(ngModle)] which means read and write in the same time.

```
<input type="text"  placeholder="your name"
[(ngModel)]="name" />

<input type="text"  placeholder="your email"
[(ngModel)]="email" />

<input type="number"  placeholder="your Salary“
[(ngModel)]="salary" />
```

And this code to read the value from typescript file .

```
<h1>Current name is : {{name}}</h1>
<h1>Current email is : {{email}}</h1>
<h1>Current salary is : {{salary}}</h1>
<h1>Current annual salary is : {{salary*12}}</h1>
```

In app.component.ts

```
export class AppComponent {
    title = 'firstProject';
    name: string = '';
    email: string = '';
    salary: number = 0;
}
```

Two way data binding

In app.component.css

```
input {
        display:  block;
        width:  300px;
        padding:  10px;
        font-size:  1em;
        margin-top:  10px;
}
```

To do the logic.

In app.component.html

```
<input type="text"placeholder="your name"  [(ngModel)]=
"name"  (ngModelChange)="handlechange($event)" />
```

In app.component.ts

```
handlechange(ev: any)
 {
      console.log(ev.length);
      if (ev.length > 15) {
          this.name = this.name.substr(0, 15);
          alert("you are writing along name  ")
      }
      if (ev.length > 20)
          alert("Stop writing !!")
 }
```

Exercise:

Add button called clear to clear all data in html page use click event.

Exercise Solution:

In app.component.html

```
<button (click) ="clearValue()"> Clear </button>
```

In app.component.ts

```
clearValue(){
  this.name = '';
  this.email = '';
  this.salary = 0;
}
```

Chapter 3

1. What is Data Binding
2. One way data binding
3. Two way data binding
4. **How to create module in angular**

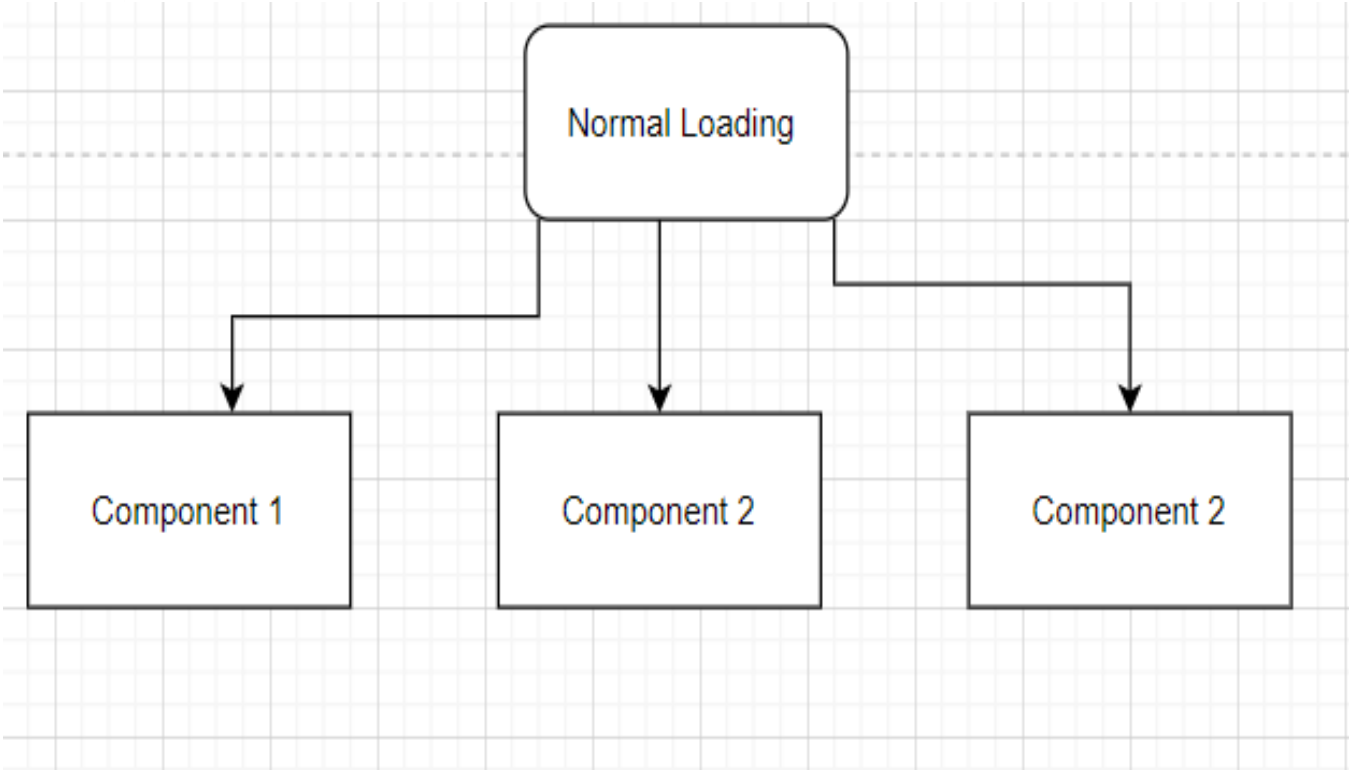
How to create module in angular

Before creating a new module, we will talk about the difference between **normal Loading** and **lazy loading.**

Normal loading

More than one component, but to call these components it must be in the same module. Like navbar and footer.

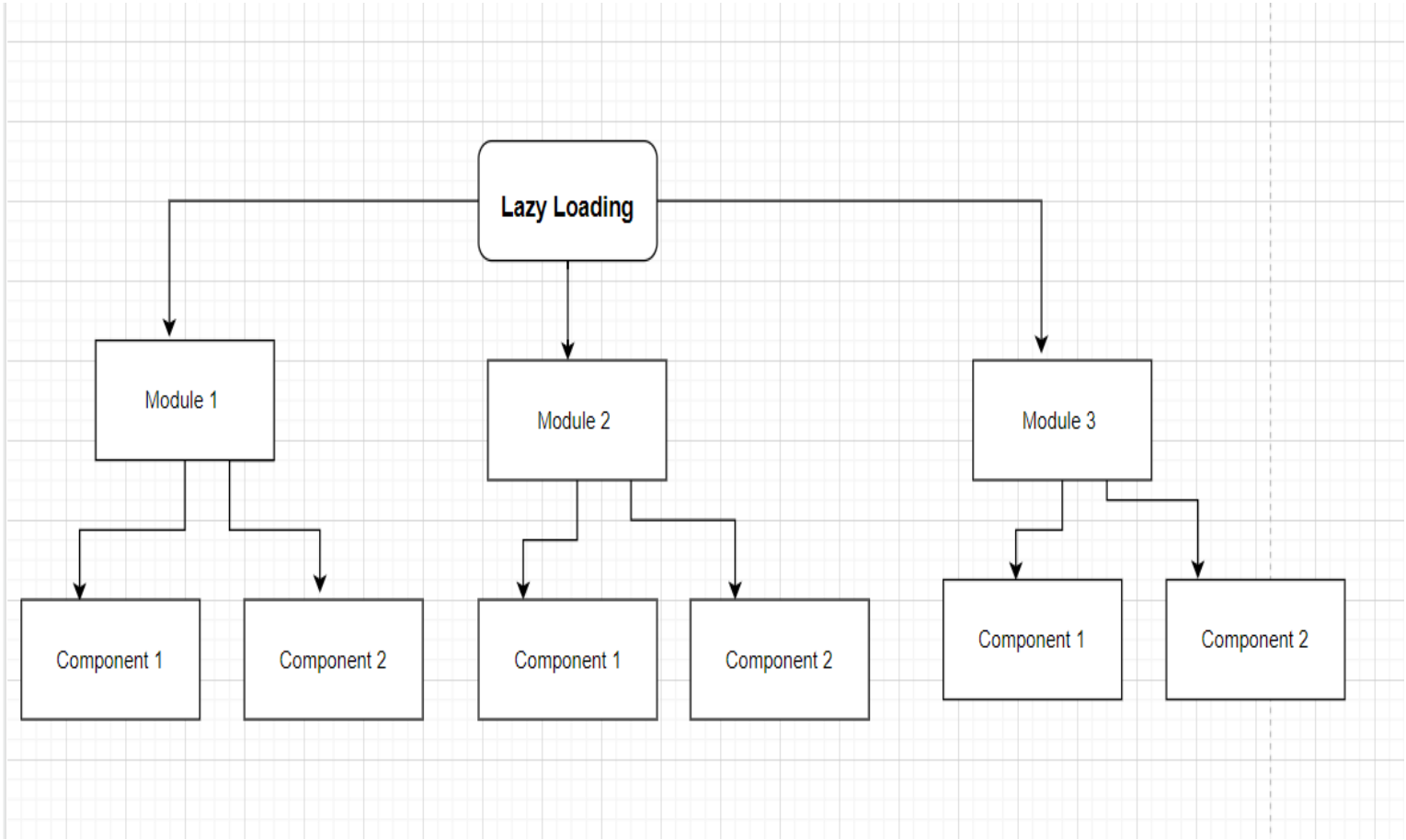
How to create module in angular



Lazy loading

It means more than one module and each module have their components and you can load the component when you need.

How to create module in angular



How to create module in angular

Use this command to generate new module.

ng generate module module _name **- -routing**

OR

ng g m module _name **- -routing**

How to create module in angular


```
PS C:\Users\User\Desktop\firstProject> ng g m auth --routing
CREATE src/app/auth/auth-routing.module.ts (247 bytes)
CREATE src/app/auth/auth.module.ts (272 bytes)
PS C:\Users\User\Desktop\firstProject>
```

Generate components for a specific module

شركة تحالف الإمارات للحلول التقنية ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

Before starting, Lets creates new project called **PortalApp**.

Then create a new module called **auth** and for this module generate two components:

Login and **Register**.

Generate components for a specific module

Create a new project called PortalApp.

```
PS C:\Users\User\Desktop> ng new PortalApp
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE PortalApp/angular.json (3057 bytes)
CREATE PortalApp/package.json (1072 bytes)
CREATE PortalApp/README.md (1055 bytes)
CREATE PortalApp/tsconfig.json (783 bytes)
CREATE PortalApp/.editorconfig (274 bytes)
CREATE PortalApp/.gitignore (604 bytes)
CREATE PortalApp/.browserslistrc (703 bytes)
CREATE PortalApp/karma.conf.js (1426 bytes)
CREATE PortalApp/tsconfig.app.json (287 bytes)
CREATE PortalApp/tsconfig.spec.json (333 bytes)
CREATE PortalApp/src/favicon.ico (948 bytes)
```

Generate components for a specific module

Create a new module called auth.

```
PS C:\Users\User\Desktop\PortalApp> ng g m auth --routing
? Would you like to share anonymous usage data about this project with the Angular Team at
Google under Google's Privacy Policy at https://policies.google.com/privacy? For more
details and how to change this setting, see https://angular.io/analytics. Yes

Thank you for sharing anonymous usage data. Would you change your mind, the following
command will disable this feature entirely:

    ng analytics project off

CREATE src/app/auth/auth-routing.module.ts (247 bytes)
CREATE src/app/auth/auth.module.ts (272 bytes)
PS C:\Users\User\Desktop\PortalApp>
```

Generate components for a specific module

Create login component in auth module. To determents these components for this module you must use moduleName/componentsName.

```
PS C:\Users\User\Desktop\PortalApp> ng g c auth/login
CREATE src/app/auth/login/login.component.html (20 bytes)
CREATE src/app/auth/login/login.component.spec.ts (619 bytes)
CREATE src/app/auth/login/login.component.ts (271 bytes)
CREATE src/app/auth/login/login.component.css (0 bytes)
UPDATE src/app/auth/auth.module.ts (352 bytes)
PS C:\Users\User\Desktop\PortalApp>
```

Generate components for a specific module

Create a register component.

```
PS C:\Users\User\Desktop\PortalApp> ng g c auth/Register
CREATE src/app/auth/register/register.component.html (23 bytes)
CREATE src/app/auth/register/register.component.spec.ts (640 bytes)
CREATE src/app/auth/register/register.component.ts (283 bytes)
CREATE src/app/auth/register/register.component.css (0 bytes)
UPDATE src/app/auth/auth.module.ts (442 bytes)
PS C:\Users\User\Desktop\PortalApp>
```