1604C054 – Hybrid Mobile Programming

Ionic UI Components 2 - Angular Service & Form Component

WEEK 06

Informatics Engineering
Universitas Surabaya

Outline

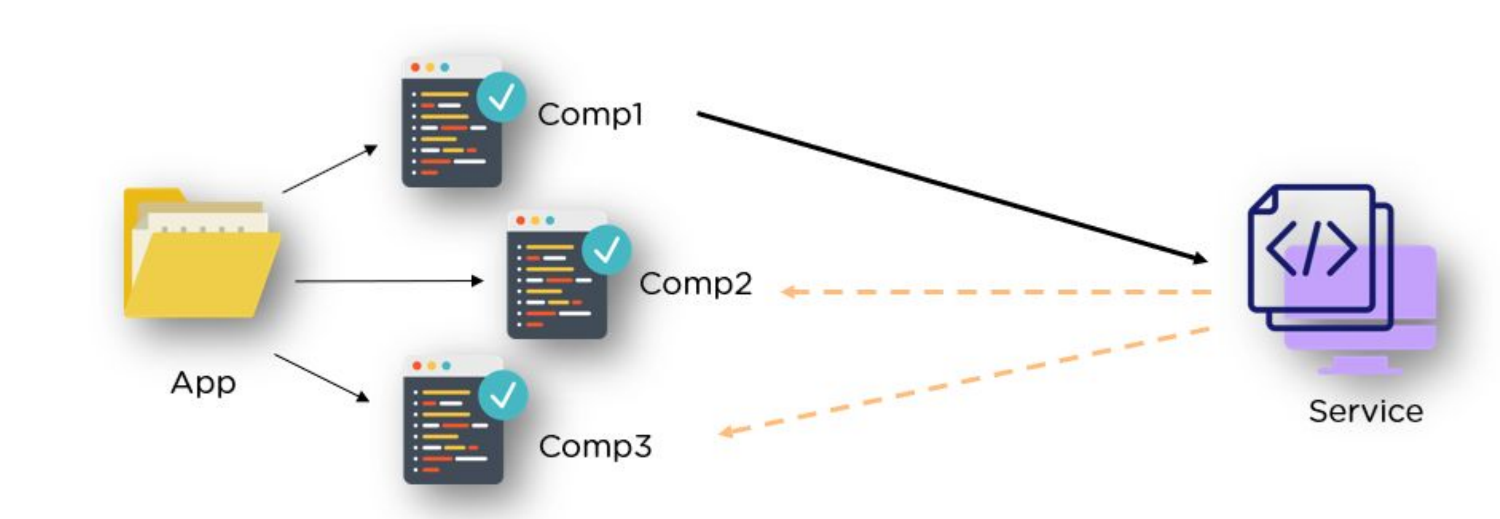
1

Angular Service

2

Form Component

Angular Service



Angular Service

TypeScript class that follows a specific pattern and is used to provide functionality that can be shared across multiple parts of an Angular application. Services are a fundamental building block of Angular applications and are typically responsible for tasks such as fetching data from a server, performing data transformations, handling application logic, and more.

Remember pastas array that redundantly placed in pasta page and pastedetail page?
Yess.. we will move this array into angular service.

Angular Service (2)

Generating angular service:

```
ionic generate service foodservice
```

Open foodservice.service.ts

paste pastas array here, delete it from pasta page.

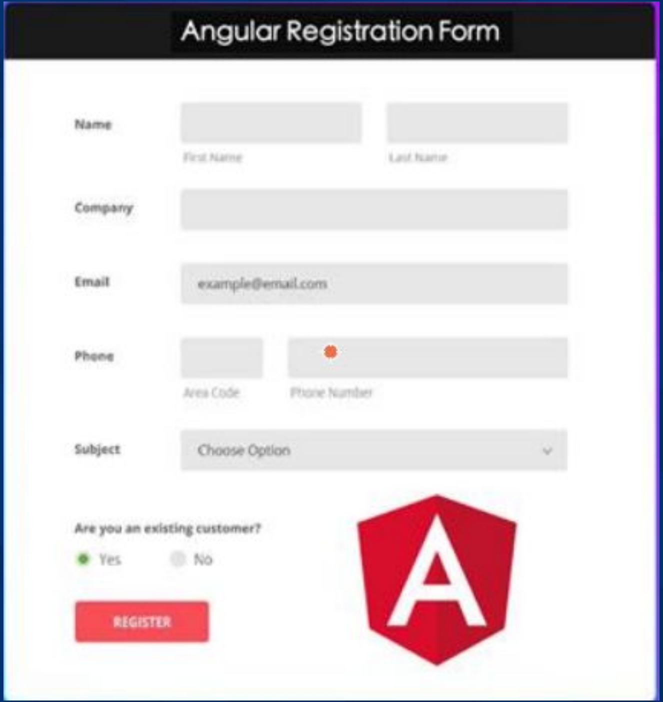
Fix pasta page.ts

```
import { FoodserviceService } from '../foodservice.service';
---
pastas:any[]=[]
constructor(private foodservice: FoodserviceService) {}

ngOnInit() {
    this.pastas=this.foodservice.pastas
}
```

service initiation

Check whether pasta page still running well.
Do similar steps in pastadetail .

Form Registration



Angular Registration Form

Name

First Name

Last Name

Company

Email example@email.com

Phone

Area Code

Phone Number

Subject Choose Option

Are you an existing customer?

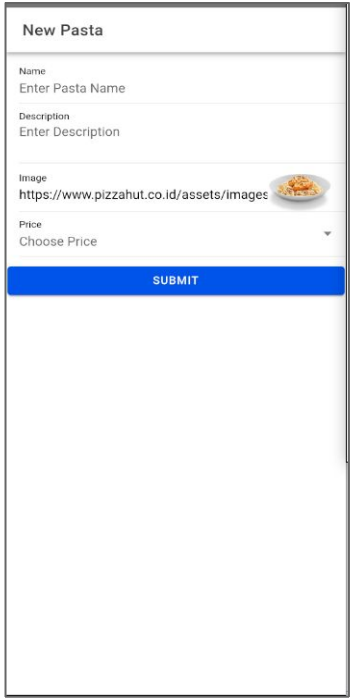
● Yes ○ No

Preparation

We will add new page called **newpasta** which will add pastas array with new member. Prepare the page and menu in the drawer that calling it.

In **page.html** we will add :

- ion-input for pasta name (property : new_name)
- ion-textarea for pasta description (property : new_desc)
- ion-select for pasta price. put options from 30000 to 50000 step 2000 (property : new_price)
- another ion-input for pasta url . you can also add ion-img to view the image (property : new_url)
- ion-button to submit above data
- all item arranged in ion-list and ion-item



New Pasta

Name
Enter Pasta Name

Description
Enter Description

Image
https://www.pizzahut.co.id/assets/images

Price
Choose Price

SUBMIT

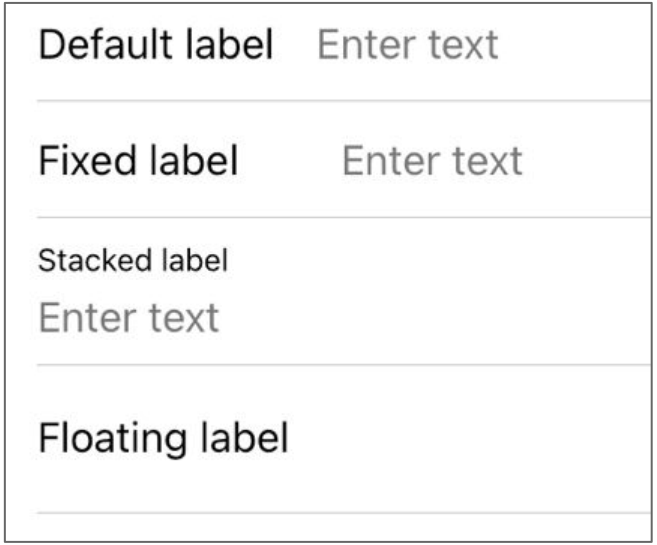
ion-input

The input component is a wrapper to the HTML input element with custom styling and additional functionality. It accepts most of the same properties as the HTML input, but works great on desktop devices and integrates with the keyboard on mobile devices.

Frequently used property:

- **type**: "text", "password", "email", "number", "search", "tel", and "url"
- **label**: used for plaintext labels
- **labelPlacement**: “fixed”, “stacked”, “floating”

Add ion input equipped with ngModel for handling pasta name!!!



Default label Enter text

Fixed label Enter text

Stacked label
Enter text

Floating label

ion-text-area

The textarea component is used for multi-line text input. A native textarea element is rendered inside of the component. The user experience and interactivity of the textarea component is improved by having control over the native textarea. Unlike the native textarea element, the Ionic textarea does not support loading its value from the inner content. The textarea value should be set in the value attribute.

Frequently used property:

- **rows**: The number of visible text lines for the control.
- **auto-grow**: If true, the textarea container will grow and shrink based on the contents of the textarea.

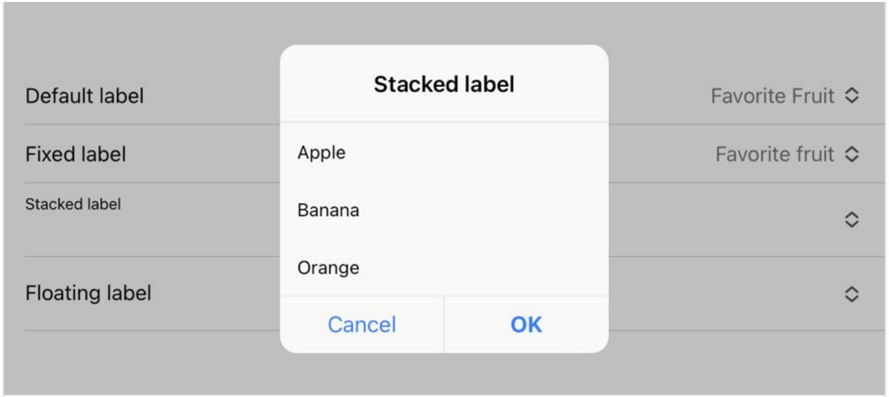
Add ion-text-area
equipped with ngModel
for handling pasta desc!!!

ion-select

Selects are form controls to select an option, or options, from a set of options, similar to a native <select> element. When a user taps the select, a dialog appears with all of the options in a large, easy to select list.

A select should be used with child <ion-select-option> elements. If the child option is not given a value attribute then its text will be used as the value.

If value is set on the <ion-select>, the selected option will be chosen based on that value.



Default label — Favorite Fruit

Fixed label — Favorite fruit

Stacked label

Floating label

Stacked label

Apple

Banana

Orange

Populating number array

Create function at page.ts

```
arr_price:number[] = []
---
ngOnInit() {
     this.arr_price=this.generateNumberOptions(30000,50000,2000)
}

generateNumberOptions(start: number, end: number, step: number): number[] {
    const options: number[] = [];
    for (let i = start; i <= end; i += step) {
      options.push(i);
    }
    return options;
}
```

```
<ion-select [(ngModel)]="new_price" label="Price" labelPlacement="floating" >
     <ion-select-option *ngFor="let price of arr_price"
            [value]="price">{{price}}</ion-select-option>
</ion-select>
```

ion-button

Buttons provide a clickable element, which can be used in forms, or anywhere that needs simple, standard button functionality. They may display text, icons, or both. Buttons can be styled with several attributes to look a specific way.

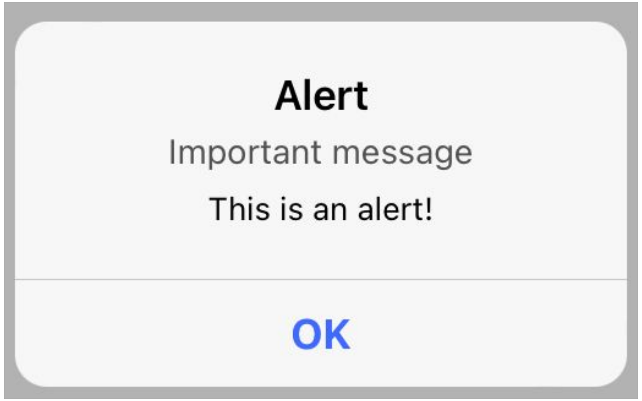
Frequently used property:

- **expand:** Set to "block" for a full-width button or to "full" for a full-width button with square corners and no left or right borders.
- **fill:** Set to "clear" for a transparent button that resembles a flat button, to "outline" for a transparent button with a border, or to "solid" for a button with a filled background. The default fill is "solid" except inside of a toolbar, where the default is "clear".
- **shape:** Set to "round" for a button with more rounded corners.
- **size:** “default” | “large” | “small”
- **color:** “primary” | “secondary” | “tertiary” | “sucess” etc

ion-alert

An Alert is a dialog that presents users with information or collects information from the user using inputs. An alert appears on top of the app's content, and must be manually dismissed by the user before they can resume interaction with the app. It can also optionally have a header, subHeader and message.

```
<ion-button id="present-alert">Click Me</ion-button>
<ion-alert
  trigger="present-alert"
  header="Alert"
  subHeader="Important message"
  message="This is an alert!"
  [buttons]="alertButtons"
></ion-alert>
```


Alert

Important message

This is an alert!

in page.ts

```
public alertButtons = ['OK']
```

Add new member to pastas

Usually, an angular service is responsible to operate CRUD using web service. In this case, we will put a function in angular service to store another pasta data to pastas array.

add in service.ts

```
addPasta(p_name:string,p_url:string,p_description:string,p_price:number)
{
  this.pastas.push({name:p_name, url:p_url,
    description:p_description,price:p_price})
}
```

Prepare newpasta page to be ready to use foodservice. Add the import and initiate in constructor.

Add new member to pastas (2)

Add in newpasta.html

```
<ion-item>
      <ion-button id="btnsubmit" (click)="submitpasta()">Submit</ion-button>
      <ion-alert
        trigger="btnsubmit"
        header="Add Pasta"
        message="Data Succesfully stored"
        [buttons]="alertButtons"
      ></ion-alert>
</ion-item>
```

After submit new data,
check in pasta page, you should
see your new pasta in there.

And in newpasta.ts

```
submitpasta()
{
this.foodservice.addPasta(this.new_name,this.new_url,this.new_desc,this.new_price)
}
```

ion-avatar

Avatars are circular components that usually wrap an image or icon. They can be used to represent a person or an object.

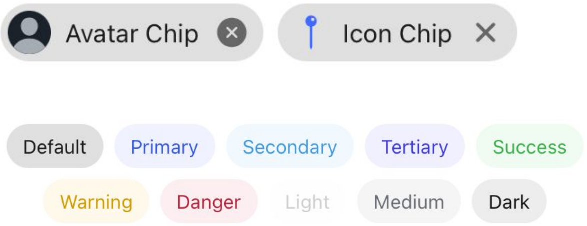
Avatars can be used by themselves or inside of any element. If placed inside of an ion-chip or ion-item, the avatar will resize to fit the parent component. To position an avatar on the left or right side of an item, set the slot to start or end, respectively.

```
<ion-avatar>
  <img alt="Silhouette of a person's head"
src="https://ionicframework.com/docs/img/demos/avatar.svg" />
</ion-avatar>
```

ion-chip

Chips represent complex entities in small blocks, such as a contact. A chip can contain several different elements such as avatars, text, and icons.

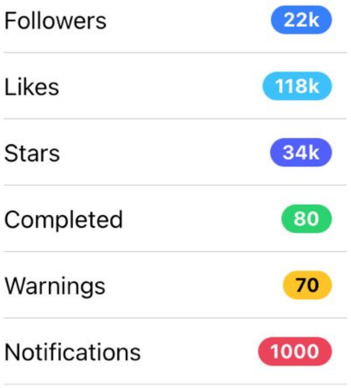
```
<ion-chip>
  <ion-avatar>
    <img alt="Silhouette of a person's head"
src="https://ionicframework.com/docs/img/demos/avatar.sv
g" />
  </ion-avatar>
  <ion-label>Avatar Chip</ion-label>
  <ion-icon name="close-circle"></ion-icon>
</ion-chip>

<ion-chip>
  <ion-icon name="pin" color="primary"></ion-icon>
  <ion-label>Icon Chip</ion-label>
  <ion-icon name="close"></ion-icon>
</ion-chip>
```



ion-badge

Badges are inline block elements that usually appear near another element. Typically they contain a number or other characters. They can be used as a notification that there are additional items associated with an element and indicate how many items there are. Badges are hidden if no content is passed in.

```
<ion-list>
  <ion-item>
    <ion-label>Followers</ion-label>
    <ion-badge color="primary">22k</ion-badge>
  </ion-item>
  <ion-item>
    <ion-label>Likes</ion-label>
    <ion-badge color="secondary">118k</ion-badge>
  </ion-item>
  <ion-item>
    <ion-label>Stars</ion-label>
    <ion-badge color="tertiary">34k</ion-badge>
  </ion-item>
  <ion-item>
    <ion-label>Completed</ion-label>
    <ion-badge color="success">80</ion-badge>
  </ion-item>
  <ion-item>
    <ion-label>Warnings</ion-label>
    <ion-badge color="warning">70</ion-badge>
  </ion-item>
  <ion-item>
    <ion-label>Notifications</ion-label>
    <ion-badge color="danger">1000</ion-badge>
  </ion-item>
</ion-list>
```



Followers 22k

Likes 118k

Stars 34k

Completed 80

Warnings 70

Notifications 1000

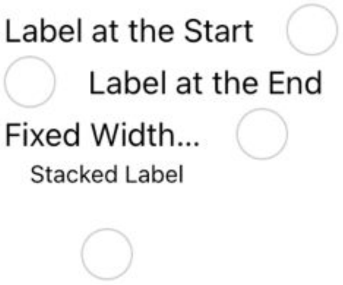
Exercise 1 (max 10 minutes)

Add in each pasta array member with spicy field. It have a boolean data type, value is true when the pasta is spicy. Modify the pasta display, add with ion-badge or ion-chip as a sign if the pasta is spicy.

ion-checkbox

Checkboxes allow the selection of multiple options from a set of options. They appear as checked (ticked) when activated. Clicking on a checkbox will toggle the checked property. They can also be checked programmatically by setting the checked property.

```
<ion-checkbox labelPlacement="start">Label at the
Start</ion-checkbox>
<br />
<ion-checkbox labelPlacement="end">Label at the
End</ion-checkbox>
<br />
<ion-checkbox labelPlacement="fixed">Fixed Width
Label</ion-checkbox>
<br />
<ion-checkbox labelPlacement="stacked">Stacked
Label</ion-checkbox>
```



Exercise 2 (max 10 minutes)

Add newpasta page with ion-checkbox to handle the spicy field.

Exercise 2 (max 10 minutes)

Add an avatar ion which contains an ion-img wrapped in a div

```
<div id="frame">
  <ion-avatar>
   <ion-img src="https://i.pravatar.cc/300?img=15"></ion-img>
  </ion-avatar>
</div>
```

You can see that there is already a photo in the circle, yes but not too big as desired, and is on the left. Open about.page.scss , add :

```
ion-avatar{
width: 120px;
height:120px;
}
#frame{
display: flex;
justify-content: center;
margin-top:20px;
}
```

Redirecting Page

In Ionic Angular, you can use the Router service provided by Angular's Router module (@angular/router) to navigate between pages or components.

add import:

```
import { Router } from '@angular/router';
```

add injection:

```
constructor(private router: Router) {}
```

use it:

```
this.router.navigate(['/target-route']);
```

Exercise 3 (max 5 minutes)

Right after submitting new pasta,redirect to pasta list page.

Thanks.

Any Question ?