



ANGULAR 2 FUNDAMENTALS

December 13, 2017

PART 7

INTRODUCING SERVICES

- App is running well but there are 2 problems
 - Creating images in the image-list component is not the right way if doing it.
 - What if we want to share the images with other components or views

SERVICES

- In angular
 - We may create a service and use it to manage our list of images
 - The service can also be used across all components

LET'S CREATE A SERVICE

- `cd ../`
- `mkdir services`
- `cd services`
- `ng s image`

- `//service` is generated but not provided

REGISTER THE SERVICE

- Open `app.module.ts`
- In `providers:[]`;
- add:
 - `providers: [ImageService],`
- don't forget to import `ImageService` at the top of file:

```
import {ImageService} from './services/image.service';
```


IMAGE.SERVICE.TS

.....

```
import { Injectable } from '@angular/core';
import { Image } from '../models/image';

@Injectable()
export class ImageService {

  images: Image[] = [
    new Image('1', 'First image', 'First image description', 'https://angularbooks.com/img/angular4/img1.jpg', 'https://angularbooks.com/img/angular4/img1-1.jpg'),
    new Image('2', 'Second image', 'Second image description', 'https://angularbooks.com/img/angular4/img2.jpg', 'https://angularbooks.com/img/angular4/img2-1.jpg'),
    new Image('3', 'Third image', 'Third image description', 'https://angularbooks.com/img/angular4/img3.jpg', 'https://angularbooks.com/img/angular4/img3-1.jpg'),
    new Image('4', 'Fourth image', 'Fourth image description', 'https://angularbooks.com/img/angular4/img4.jpg', 'https://angularbooks.com/img/angular4/img4-1.jpg'),
    new Image('5', 'Fifth image', 'Fifth image description', 'https://angularbooks.com/img/angular4/img5.jpg', 'https://angularbooks.com/img/angular4/img5-1.jpg'),
    new Image('6', 'Sixth image', 'Sixth image description', 'https://angularbooks.com/img/angular4/img6.jpg', 'https://angularbooks.com/img/angular4/img6-1.jpg'),
    new Image('7', 'Seventh image', 'Seventh image description', 'https://angularbooks.com/img/angular4/img7.jpg', 'https://angularbooks.com/img/angular4/img7-1.jpg'),
    new Image('8', 'Eighth image', 'Eighth image description', 'https://angularbooks.com/img/angular4/img8.jpg', 'https://angularbooks.com/img/angular4/img8-1.jpg'),
  ];
}
```

IMAGE.SERVICE.TS

```
constructor() { }  
  
getImages() {  
    return this.images;  
}  
}
```

Adding the @injectable decorator at the top makes us use the service in other components

USING THE SERVICE

- Open image-list.component.ts find:

```
import { Component, OnInit } from '@angular/core';  
import { Image } from '../models/image';
```

- Add below:

```
import { ImageService } from '../services/image.service';
```

- Inject image service in the constructor:

```
constructor(private imageService: ImageService) { }
```

- Create empty array of images:

```
images: Image[] = [];
```

USING SERVICE

- Call service and get all images

```
ngOnInit() {  
  this.images = this.imageService.getImages();  
}
```

IMAGE-LIST.COMPONENT.TS

```
import { Component, OnInit } from '@angular/core';
import { Image } from '../../models/image';
import { ImageService } from '../../services/image.service';

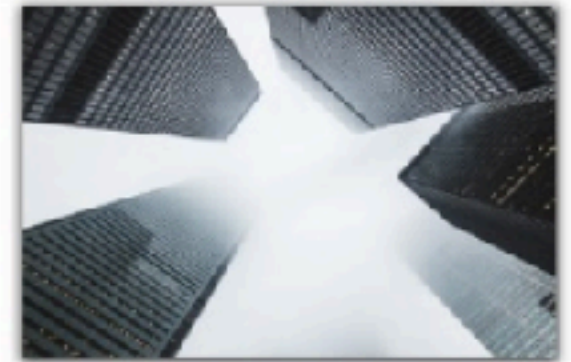
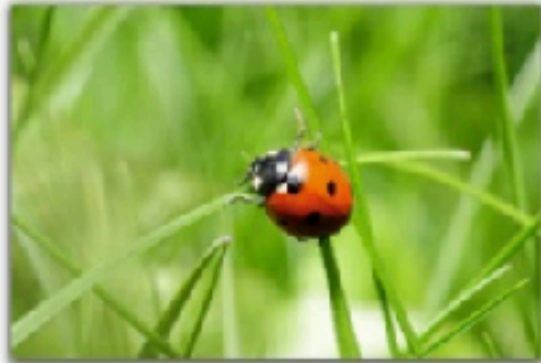
@Component({
  selector: 'ng-image-list',
  templateUrl: './image-list.component.html',
  styles: []
})
export class ImageListComponent implements OnInit {
  images: Image[] = [];

  constructor(private imageService: ImageService) { }

  ngOnInit() {
    this.images = this.imageService.getImages();
  }
}
```

CHECK APP AGAIN

.....



LIFECYCLE HOOKS

constructor: The constructor is always called first.

ngOnInit: This lifecycle hook is called after the constructor.

ngOnChanges: When the value of an input is changed, this lifecycle hook is executed.

ngDoCheck: When the input properties are checked, this method will be called.

ngAfterContentInit: When the content of the component is initialized, this hook is called. A component-only hook.

ngAfterContentChecked: Called after the content is checked. A component-only hook.

ngAfterViewInit: After a view is initialized, this hook is called. A component-only hook.

ngAfterViewChecked: After a view is checked, this hook is called. A component-only hook.

ngOnDestroy: This method is called before Angular destroy the component/directive.

END OF PART 7