Angular JS 2.0

XORIANT

Contents

Module	Topic
Module 1	Observables / RxJS
Module 2	Routes

Angular communication with server

There are 3 ways to handle REST call:

- ***Callbacks:*** Handling REST call using callback function is very preliminary way of communication. Once REST call sends response, suitable success / error function is called where developer can take certain action on response.
- ***Promises:*** A promise represents a value that we can handle at some point in the future. Promises allow to execute the action multiple times for a single event & hence Promises are preferred over Callbacks.
- ***Observables:*** Observables open up a continuous channel of communication in which multiple values of data can be emitted over time. Every observable is a promise plus advance features.

Angular 2 support for REST communication

Angular 2 provides a separate module called ‘HttpModule’ in order to communicate with server. Here are the steps to include http support in Angular application:

1. Include HttpModule into app.module.ts

```
import { HttpModule } from '@angular/http';
imports: [ BrowserModule, HttpModule ],
```

2. Inject Http service in your service class

```
import { Http, Response } from '@angular/http';
export class ProductService {
constructor(private _http: Http) {
}
}
```

3. Invoke REST call using http service: *this._http.get(‘http://localhost:8000/product’);*

What is an Observable?

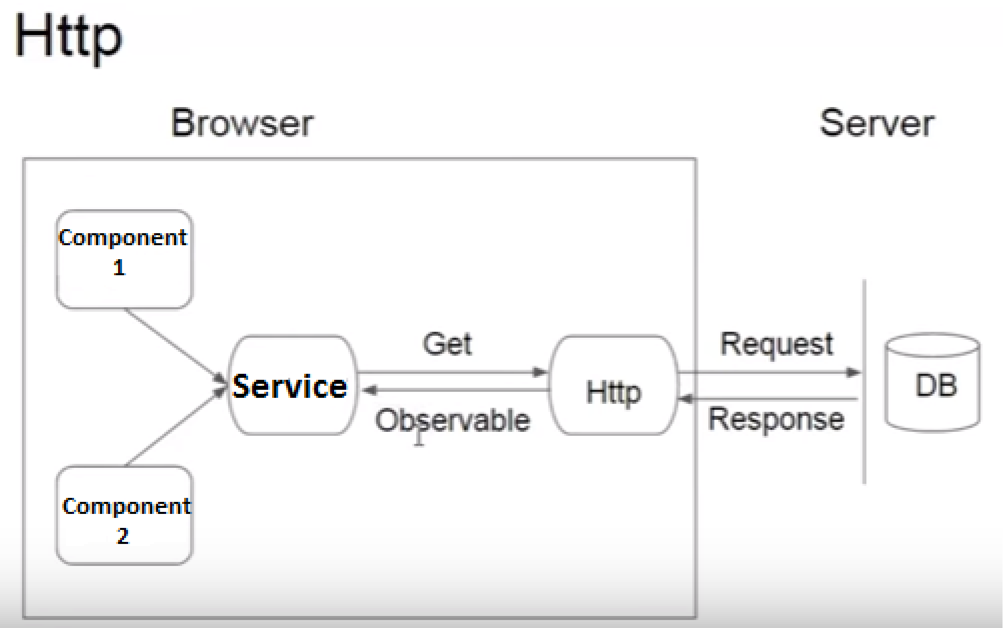
- Observable isn't an Angular specific feature, but rather a proposed standard for managing async data that will be included in the release of ES7.
- Observable is a sequence of items that arrive async over the time. However, with ‘http’ service calls it is always a single item also known as http_response.
- Since, Observable feature is not available in ES6 specification, we use Observable from a third party library called RxJS.

RxJS

- RxJS stands for Reactive Extensions for JavaScript.
- Observable feature is not available in ES6 specification. Hence, we use Observable from a third party library called RxJS.
- In order to download RxJS in angular application, just add its dependency into package.json

```
"dependencies": {
        "rxjs": "5.0.1"
}
```

Using Observable in Angular 2 App



Steps to use Observable in Angular App:

1. Make http call from Service.
2. Receive observable & map it.
3. Subscribe to observable in your components.
4. Render the view using received data.

Using Observable in Angular App

ProductService.ts

```
import { Http, Response } from '@angular/http';
import 'rxjs/add/operator/map';

@Injectable()
export class ProductService {
        constructor(private _http: Http) {
        }
        getProducts() {
                return this._http.get(this.url).
                        map((response: Response)=>response.json());
        }
}
```

Using Observable in Angular App continue…

ProductComponent.ts

```
import { ProductService } from './products.service';
@Component({
    providers: [ProductService]
})
export class ProductComponent {
constructor(productService: ProductService) {
    productService.getProducts().subscribe((data)=> {
            this.products=data;
            console.log(‘Received products: ', this.products)
    });
}
```

Using Observable for POST call

ProductService.ts

```
createProduct(product: Product) {
        let headers = new Headers({ 'Content-Type': 'application/json' });
        let options = new RequestOptions({ headers: headers });
        return this._http.post(this.url, JSON.stringify(product), options)
                      .map((response: Response)=>response.json());
}
```

Promises vs Observable

Sr. No.	Promise	Observable
1.	Promise cannot be cancelled.	Observable can be cancelled.
2.	Promise cannot be retried.	Observables can be retried using retry() or retryWhen() functions.
3.	Primise is a request with single return value.	Observable is a request that can return multiple response as an async stream.

What is a Router?

An Angular application is a collection of multiple components & you need to switch from one component to another based upon action performed by end user. Thus, Angular Router will help to navigate from one angular compone to another.

Steps to introduce Router in Angular App

- Set the <base> tag into index.html. It will help your application to understand how to construct url's while navigation.

```
<head>
        <base href="/">
```

- Import RouterModule into AppModule & mention routing details.

```
import { RouterModule } from '@angular/router';
imports: [ BrowserModule,
        RouterModule.forRoot([
                {path: 'first', component: FirstComponent},
                {path: 'second', component: SecondComponent}
        ])
```

Steps to introduce Router in Angular App continue…

- Add ‘active’ class in styles.css

```
nav a.active {
        color: orange;
}
```

- Finally in the navigation component, provide the links & specify router outlet to render the required component.

```
<nav>
        <a routerLink="/first" routerLinkActive="active" >First</a>
        <a routerLink="/second" routerLinkActive="active" >Second</a>
</nav>
<router-outlet></router-outlet>
```

Route parameters

When navigating from one to other component, the current component may wish to send few parameters to another component. Here are the steps to make it possible:

- Register route url with parameter into app.module.ts

```
RouterModule.forRoot([
        {path: 'fourth/:name', component: FourthComponent}        ])
```

- Pass the parameter while source component is navigating to target component:

```
import { Router } from '@angular/router';
export class ThirdComponent {
        constructor(private router: Router) {}
        onClick(){   this.router.navigate(['/fourth', 'Anand']);      }
}
```

Route parameters continue…

Finally, read the supplied parameter into target component:

```
import { ActivatedRoute } from '@angular/router';
@Component({
        selector: '<fourth-comp>',
        template: '<h3>Fourth Component says Hello {{name}}</h3>'
})
export class FourthComponent implements OnInit{
        name: string;
        constructor(private activatedRoute: ActivatedRoute) {}

        ngOnInit() {
                this.name = this.activatedRoute.snapshot.params['name'];
        }
}
```

Thank You!

US – Corporate Headquarters

1248 Reamwood Avenue,
Sunnyvale, CA 94089
Phone: (408) 743 4400

343 Thornall St 720
Edison, NJ 08837
Phone: (732) 395 6900

UK

20 Broadwick Street
Soho, London
W1F 8HT, UK

89 Worship Street
Shoreditch,
London EC2A 2BF, UK
Phone: (44) 2079 938 955

India

Mumbai
4th Floor, Nomura
Powai , Mumbai 400 076

Pune
5th Floor, Amar Paradigm
Baner, Pune 411 045

Kolkata
2B, 12th Floor, Tower ‘C’
Rajarhat, Kolkata 700 156

Bangalore
4th Floor, Kabra Excelsior,
80 Feet Main Road,
Koramangala 1st Block,
Bengaluru (Bangalore) 560034

Gurgaon
A/373rd Floor, Sigma Center
Gurgaon, Haryana 122 011s