Exercise (Instructions): Angular and REST

Objectives and Outcomes

In this exercise you will learn about the a third-party Angular library called Ngx-Restangular that enables acccess to a RESTful server. Restangular simplifies common GET, POST, DELETE, and UPDATE requests with a minimum of client code. It's a perfect fit for any WebApp that consumes data from a RESTful API. At the end of this lesson you will be able to:

- Enable your Angular application to use Ng2-Restangular
- Make use of Ngx-Restangular to access a server that supports a RESTful API.

Adding and Configuring Ngx-Restangular

- First install Ngx-Restangular into your Angular application using NPM as follows:

```
npm install --save ngx-restangular
```

- Add a file named restConfig.ts to the shared folder and update its contents as follows:

```
import { baseURL } from './baseurl';

// Function for settting the default restangular configuration
export function RestangularConfigFactory (RestangularProvider) {
  RestangularProvider.setBaseUrl(baseURL);
}
```

Update AppModule to use Ng2-Restangular

- Open app.module.ts and update it as follows:

```
. . .
import { RestangularModule, Restangular } from 'ngx-restangular';
import { RestangularConfigFactory } from './shared/restConfig';

. . .

imports: [
  . . .
  RestangularModule.forRoot(RestangularConfigFactory)
  ]
  . . .
```

Updating Dish Service

- Open dish.service.ts and update it as follows to make it use ng2-restangular:

```
. . .
import { RestangularModule, Restangular } from 'ngx-restangular';

. . .
  constructor(private restangular: Restangular,
              private processHTTPMsg: ProcessHTTPMsg) { }

  getDishes(): Observable<Dish[]> {
    return this.restangular.all('dishes').getList();
  }

  getDish(id: number): Observable<Dish> {
    return  this.restangular.one('dishes',id).get();
  }

  getFeaturedDish(): Observable<Dish> {
    return this.restangular.all('dishes').getList({featured: true})
      .map(dishes => dishes[0]);
  }

  getDishIds(): Observable<number[]> {
    return this.getDishes()
      .map(dishes => { return dishes.map(dish => dish.id) })
      .catch(error => { return error; } );
  }
  . . .
```

- Save all the changes and do a Git commit with the message "REST".

Conclusions

In this exercise you updated the Angular application to use ng2-restangular to access a server supporting RESTful API.

Mark as completed