Angular 2 Tutorial

- Angular 2 - Home
- Angular 2 - Overview
- Angular 2 - Environment
- Angular 2 - Hello World
- Angular 2 - Modules
- Angular 2 - Architecture
- Angular 2 - Components
- Angular 2 - Templates

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

- Angular 2 - Directives
- Angular 2 - Metadata
- Angular 2 - Data Binding
- CRUD Operations Using HTTP
- Angular 2 - Error Handling
- Angular 2 - Routing
- Angular 2 - Navigation
- Angular 2 - Forms
- Angular 2 - CLI
- Angular 2 - Dependency Injection
- Angular 2 - Advanced Configuration
- Angular 2 - Third Party Controls
- Angular 2 - Data Display

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

- Angular 2 - Handling Events
- Angular 2 - Transforming Data
- Angular 2 - Custom Pipes
- Angular 2 - User Input
- Angular 2 - Lifecycle Hooks
- Angular 2 - Nested Containers
- Angular 2 - Services

Angular 2 Useful Resources

- Angular 2 - Quick Guide
- Angular 2 - Useful Resources
- Angular 2 - Discussion

Angular 2 - Error Handling

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Advertisements

AGILE MANIFESTO

Agile Security Manifesto Adds Security Priciples to Help Mitigate ...

Previous Page Next Page

Angular 2 applications have the option of error handling. This is done by including the ReactJS catch library and then using the catch function.

Let’s see the code required for error handling. This code can be added on top of the chapter for CRUD operations using http.

In the product.service.ts file, enter the following code −

```
import { Injectable } from '@angular/core';
import { Http , Response } from '@angular/http';
import { Observable } from 'rxjs/Observable';

import 'rxjs/add/operator/map';
import 'rxjs/add/operator/do';
import 'rxjs/add/operator/catch';
import { IProduct } from './product';

@Injectable()
export class ProductService {
   private _producturl = 'app/products.json';
   constructor(private _http: Http){}

   getproducts(): Observable<IProduct[]> {
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
      return this._http.get(this._producturl)
      .map((response: Response) => <IProduct[]> response.json())
      .do(data => console.log(JSON.stringify(data)))
      .catch(this.handleError);
   }
   private handleError(error: Response) {
      console.error(error);
      return Observable.throw(error.json().error());
   }
}
```

- The catch function contains a link to the Error Handler function.
- In the error handler function, we send the error to the console. We also throw the error back to the main program so that the execution can continue.

Now, whenever you get an error it will be redirected to the error console of the browser.

⊖ Previous Page
Next Page ⊕

Advertisements

AGILE MANIFESTO

Synopsys

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Agile Security Manifesto Adds Security Priciples to Help Mitigate Security Risks

© Copyright 2017. All Rights Reserved.

Enter email for newslette go

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com