# Angular Best Friends

## Module 8 Exercise 2 – Http errors

## Goal

Sometimes things don’t go the right way and the Http client may return errors. A proper Angular application should take this into consideration and implement error handling. In this exercise we’ll do precisely this.

## Steps

### Setting up the environment

1. Go to the **Module8Exercise2** folder. Open the “**initial**” folder in VS Code.
2. Open the terminal and run **npm install** to install the needed node modules.
3. This app is slightly different than all others because it also has a NodeJS server that serves the initial index.html file and that has API endpoints to play around with resources. So to build everything up run **npm start**.
4. Wait until you see the Angular specific bundles in the terminal.
5. Open a browser tab and navigate to **localhost:3000** . The NodeJs server listens to this port and this is how we run this Angular application

### Handling errors

1. Go to the **data.service.ts** file. Here we’ll want to update the getAllBooks method to return custom errors if something goes wrong with the Http request.
2. First update the RxJS specific imports with new components:

import { Observable, throwError } from 'rxjs';

import { map, tap, catchError } from 'rxjs/operators';

1. Delete the entire getAllBooks method and replace it with this one:

getAllBooks(): Observable<Book[] | BookTrackerError> {

console.log('Getting all books from the server.');

return this.http.get<Book[]>('/api/books')

.pipe(

catchError(err => this.handleHttpError(err))

);

}

1. This will cause some errors since we’re calling a handleHttpError method that we haven’t defined. So immediately after the getAllBooks method add this code snippet:

private handleHttpError(error: HttpErrorResponse): Observable<BookTrackerError> {

let dataError = new BookTrackerError();

dataError.errorNumber = 100;

dataError.message = error.statusText;

dataError.friendlyMessage = 'An error occurred retrieving data.';

return throwError(dataError);

}

1. The return type of this method is HttpErrorResponse. However this class is not imported yet. So either import it together with the other HttpClient specific classes or let VS Code add it for you.
2. Test it by refreshing the app and take a look at the networking tab of the browser developer tools.