

UNIT 2

Angular



Exercise week 9

Client-side Web Development
2nd course – DAW
IES San Vicente 2024/2025
Author: Arturo Bernal Mayordomo

Index

- Exercise.....3
 - Directives.....3
 - Interfaces.....3
 - Services.....3
 - Components.....3
 - Interceptors.....3

Exercise

Update the exercise from last week (week 8) with the following changes:

- We'll use the same web services from week 3 (Unit 1) exercise (<https://api.fullstackpro.es/svtickets-lite>).

Directives

Create the **encodeBase64** directive we've seen in class and use it in the **event-form** component (file input). Delete the code that's not needed anymore in the component

Interfaces

Create the necessary interfaces for the server responses: **EventsResponse**, **SingleEventResponse**, ...

Services

Create the **events** service to make requests to the server:

- **EventsService** → This service will manage everything related to events and have at least the following methods:
 - **getEvents(): Observable<MyEvent[]>**
 - **getEvent(id: number): Observable<MyEvent>**
 - **addEvent(event: MyEvent): Observable<MyEvent>**
 - **deleteEvent(id: number): Observable<void>**

Components

Components like **events-page**, **event-form** and **event-card** will use the **Events Service** to get data from the server, insert new events and delete events (at the server). Wait for the server's response (**subscribe**) before changing anything locally.

Cancel subscriptions from observable using **takeUntilDestroyed**. Also, remember you won't need the event's id in the **product-form** component anymore.

Interceptors

Create an interceptor called **baseUrlInterceptor** that adds the server prefix (<https://api.fullstackpro.es/assbook-lite> or <http://localhost:3000> if you execute the services locally) to the url of any http request.

Don't forget to delete the .angular folder alongside the node_modules

folder before uploading the exercise!