***Minimize API calls in angular using caching services***

**Introduction:**

Caching HTTP requests helps to optimize your application. Imagine requesting a piece of data from the server every time a request is placed, even if the data has never changed over time. This would impact your app’s performance due to the time it takes to process the data in the server and send it over when the data has never changed from previous requests.

To remove the time delay in processing the data in the server when it hasn’t changed, we need to check whether the data in the server has changed. If the data has changed, we process new data from the server. If not, we skip the processing in the server and send the previous data. This is called [caching](https://en.wikipedia.org/wiki/Cache_(computing)).

In HTTP, not all requests are cached. POST, PUT, and DELETE requests are not cached because they change the data in the server.

POST and PUT add data to the server while DELETE removes data from the server. So we need to process them every time they’re requested without caching them.

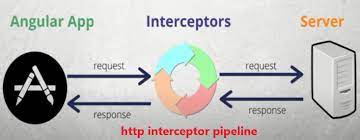
GET requests can be cached. They just get data from the server without changing them. If no POST, PUT, or DELETE request occurs before the next GET request, the data from the last GET request does not change. We simply return the previous data or response without hitting the server.

Modern browsers have a built-in mechanism to cache our HTTP requests. We’ll show you how to do this in Angular.

**Why to use HttpInterceptor?**

HttpInterceptors are special services in Angular. HTTP requests are passed through them in the chain before the actual request is made to the server.

Put simply, HttpInterceptors intercept and handle HTTP requests. Typically, HttpInterceptors call next.handle(transformedReq) to transform outgoing requests before passing them to the next interceptor in the chain. In rare cases, interceptors handle requests themselves instead of delegating to the remainder of the chain.



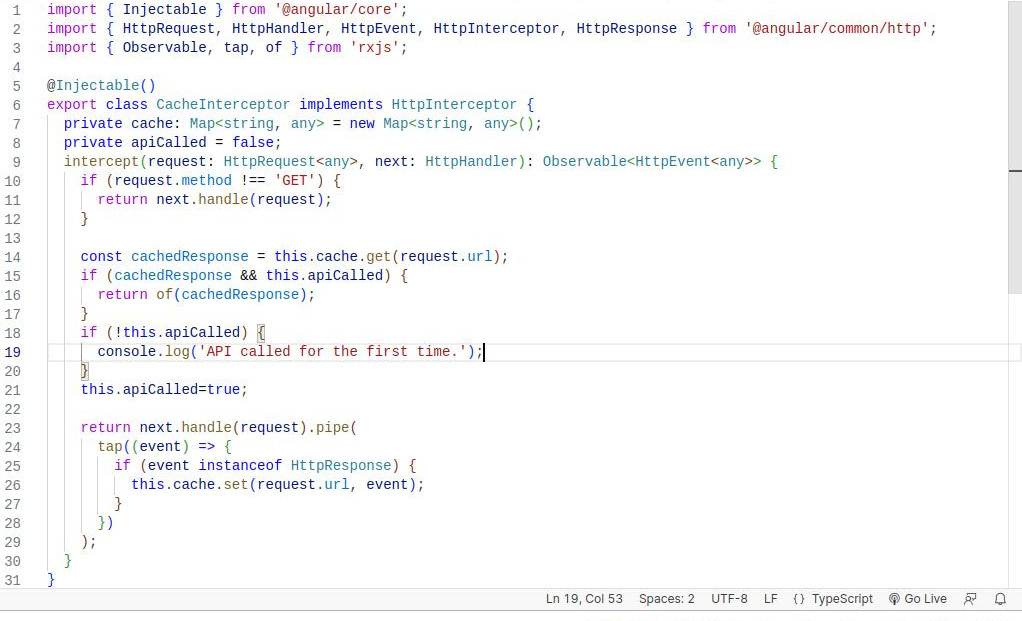
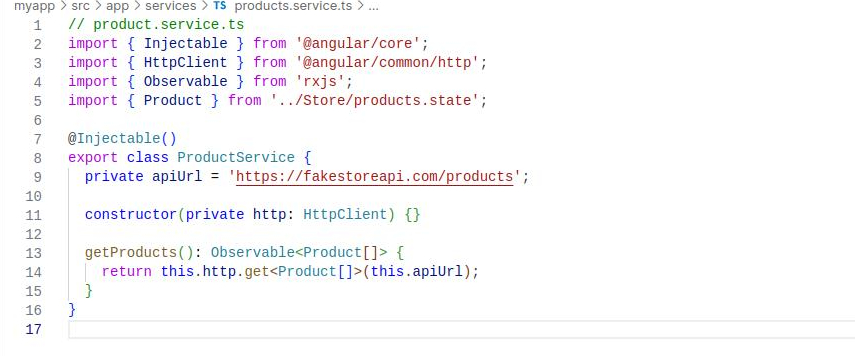
**Setting up our HttpInterceptor:**

1. Create a file “cache.interceptor.ts” in your project folder

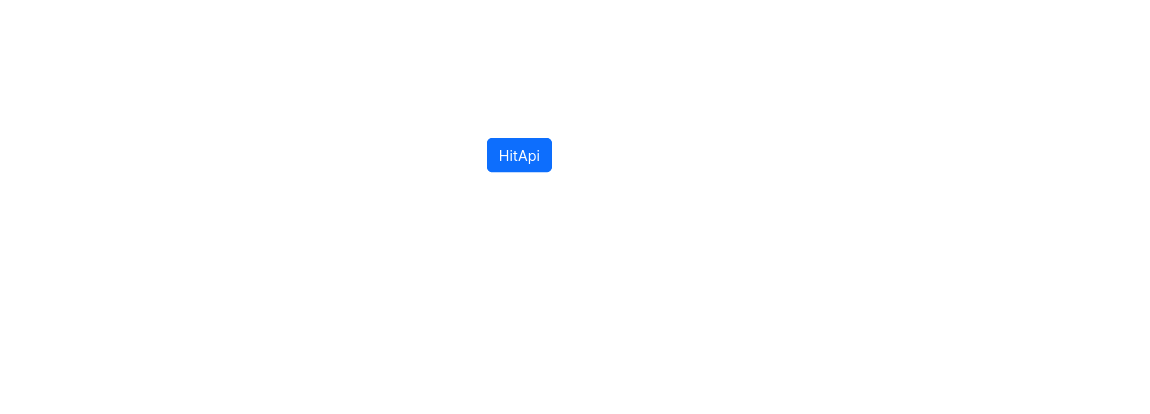
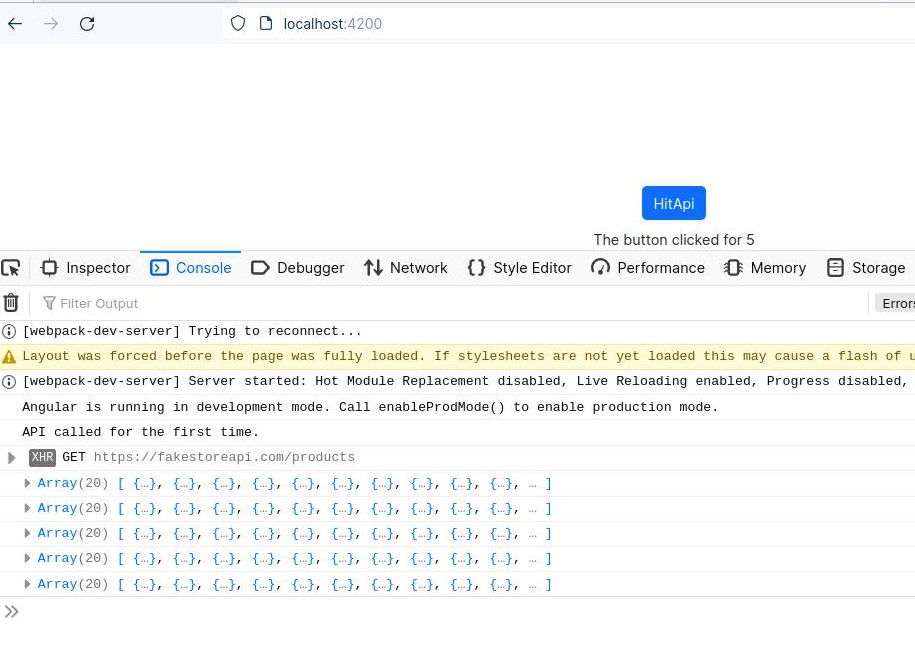
2. Add the following code to the generated interceptor file.

* Inside the intercept method, the request object is checked to determine if the request method is not GET. If it's not a GET request, the request is passed through the interceptor without caching, by calling next.handle(request).
* If the request method is GET, the interceptor checks if a response is available in the cache for the request URL. If a cached response is found and the API has been called before (apiCalled is true), the cached response is returned using of(cachedResponse).
* If there is no cached response or the API is being called for the first time, the apiCalled flag is set to true to indicate that the API has been called.
* The interceptor then calls next.handle(request) to pass the request to the next handler in the chain (such as the Angular's HTTP client).
* The interceptor intercepts the response by using the tap operator from the RxJS library. If the response is an instance of HttpResponse, it means the API call was successful.
* The successful response is then cached using this.cache.set(request.url, event) to store the response in the cache, with the request URL as the key.

**Configure the Http Interceptor:**

* Open the app.module.ts file in the src/app directory, import the HTTP\_INTERCEPTORS token, and add the interceptor to the providers array. Update the file as follows:
* After the file settings in module, Now we need to create a service file to make api request and get the response from the api.
* Generate a service with the following command.
* “ng generate service Service/product” (or) “ng g s Service/products”.
* After the service file generated add the following code.
* Now save your file and create a component of your desire naming one. Here, I’am creating with the name “home1”.
* After creating a component we will get four files in our component
* --name.component.html
* --name.component.css
* --name.component.spec.ts
* --name.component.ts
* Now navigate to the html file and desing a button of name “hitApi” and make some css styles to look it better.
* Anyway’s the code reference is provided below
* Now navigate to the “home1.component.ts” file and do subscription to get the response from the api.



* Now save your all files and run the command “ng serve” or “ng s” in your command line to run our project in the browser
* After running the above you will see a button called hit api and a screenshot for reference is provided below.
* Now If click on the button the function “hitApi()” will get activated and subscribes to the api and stores the data into the variable called responseData.
* Below is the image which shows that the api is called only once on first time but from second time onwards we will be provided with the cached data in the api.
* Image 1:
* Above the button is clicked is for 5 times but the get request is only executed once.
* Image2:
* As you can see in the above screenshot the network tab in the browser shows only one request for 5 times api calling
* So, That's it! we created an Angular project that demonstrates the use of HTTP interceptors and caching technique to minimize the api calls.

***Thank you...***

