Http Requests

Don’t receive data straight from a database send it to a server or api.

http verb: these include post, get, put, etc.

url (api endpoint): /posts/1

headers (metadata): {“content-type”:”application/json”}

body: {title: “new post”}

Add the html clientmodule to the project in order to be able to handle http requests the import is from @angular/common/http

On other apis you have to have post or add in the url.

Post returns an observable.

all requests are only sent when you subscribe

Good practice to use observable operators.

The map operators allows you to get data and return other data and turns it into an observable which allows you to subscribe.

To assign type to response type use the angled brackets on what type it is. Ex: .get<{ [key: string]: Post}> will assign to a Post method.

Try to keep all of your posts into a service so that there is as little calculating methods in the component class as possible.

You can handle errors by importing error throw from rxjs.

You can also catch errors using rxjs and catchError

Both of these functions are observables

Instead of just sending data with the request you can also send custom http objects with it like hears with custom data in them. This ca be headers, params, etc.

Params is nice because it will automatically concatenate to the url instead of having long strings to concatenate.

With multiple params you have to sore the variable in a let and set it at the top of the file and just set params equal to the variable you make. Ex: params: variableMade

Observe will give you the response of the page and allows you to look at events if you need to.

You can see the response type of the object and make sure that it keeps it as the response type you want.

The response type has to match the type of the get, post, etc. call.

Interceptors: connect headers, observe, response type to all requests. To use interceptor create a service, import http interceptor, http request, http handler, from common http and implement HttpInterceptor on the class. Remember to add the interceptor to the module.ts file.

The intercept method has a signature of req: HttpRequest<any>, nex: HttpHandler and it will return next.handle(yourData/Variable);

To add the interceptor to the module.ts file in the provide add { provide: HTTP\_INTERCEPTORS, useClass: interceptorServiceName, multi: true } into the array if you need multiple just add a comma between objects.

Rquest object is immutable.

To change objects use clone and than use the normal way of changing headers, params, etc.

You can interact with the response not just the request. You do this buy using pipe and creating an observable.

It is important the order you provide interceptors.

Headers syntax: headers: new HttpHeaders({ ‘Custom-Header’: ‘Hello’ })

Params syntax: params: new HttpParams.set(‘print’, ‘pretty’)

Observe syntax: observe: ‘response’

Response syntax: responseType: ‘text’