Listing.ts

import { NgModule, Component, Injectable } from "@angular/core";

import { BrowserModule } from "@angular/platform-browser";

import { platformBrowserDynamic } from "@angular/platform-browser-dynamic";

import { HttpClientJsonpModule, HttpClientModule, HttpClient } from "@angular/common/http";

import { ReactiveFormsModule, FormControl, FormsModule } from "@angular/forms";

import { Observable } from "rxjs";

import { map, debounceTime, distinctUntilChanged, switchMap, tap } from "rxjs/operators";

class SearchItem {

constructor(

public track: string,

public artist: string,

public link: string,

public thumbnail: string,

public artistId: string

) {}

}

@Injectable()

export class SearchService {

apiRoot: string = "https://itunes.apple.com/search";

constructor(private http: HttpClient) {}

search(term: string) {

let apiURL = `${this.apiRoot}?term=${term}&media=music&limit=20`;

return this.http.jsonp<any>(apiURL, "callback").pipe(

map(res => {

return res.results.map(item => {

return new SearchItem(

item.trackName,

item.artistName,

item.trackViewUrl,

item.artworkUrl30,

item.artistId

);

});

})

);

}

}

@Component({

selector: "app",

template: `

<form class="form-inline">

<div class="form-group">

<input type="search" class="form-control" placeholder="Enter search string" [formControl]="searchField">

</div>

</form>

<div class="text-center">

<p class="lead" \*ngIf="loading">Loading...</p>

</div>

<ul class="list-group">

<li class="list-group-item" \*ngFor="let track of results | async">

<img [src]="track.thumbnail" alt="{{track.track}}">

<a target="\_blank" [href]="track.link">{{ track.track }}</a>

</li>

</ul>

`

})

class AppComponent {

private loading: boolean = false;

private results: Observable<SearchItem[]>;

private searchField: FormControl;

constructor(private itunes: SearchService) {}

ngOnInit() {

this.searchField = new FormControl();

this.results = this.searchField.valueChanges.pipe(

debounceTime(400),

distinctUntilChanged(),

tap(\_ => (this.loading = true)),

switchMap(term => {

if (!term.trim()) {

return [];

}

return this.itunes.search(term);

}),

tap(\_ => (this.loading = false))

);

}

}

@NgModule({

imports: [

BrowserModule,

ReactiveFormsModule,

FormsModule,

HttpClientModule,

HttpClientJsonpModule

],

declarations: [AppComponent],

bootstrap: [AppComponent],

providers: [SearchService]

})

class AppModule {}

platformBrowserDynamic().bootstrapModule(AppModule);