

## TOOLS FOR PROJECT

### Google Chrome

<https://www.google.com/chrome/>

### Nodejs

<https://nodejs.org/es/>

CMD: npm -v

### TypeScript

<https://www.typescriptlang.org/>

CMD: npm install -g typescript

### AngularCLI

<https://cli.angular.io/>

CMD: npm install -g @angular/cli

- ng new my-dream-app
- cd mojix-movies-app
- ng serve

### Postman

<https://www.getpostman.com/>

### Visual Studio Code

<https://code.visualstudio.com/>

- AB Html Formatter
- Activitus Bar
- Angular Snippets
- Angular Language Service
- ionic Snippets
- Paste JSON as Code
- TypeScript Importer

### Ionic y Cordova

<https://ionicframework.com/>

CMD: npm install -g ionic

CMD: npm install -g cordova

**Android Studio**

<https://developer.android.com/studio/>

## CROC PROJECT

### Description:

This Project is a small DEMO for simulate a environment based on <https://www.themoviedb.org/>, to continue, show the steps that is part of project:

1. Creating a API KEY on <https://www.themoviedb.org/>

The screenshot shows the TMDb sign-up form. A green callout bubble points to the right side of the 'Sign up for an account' section, containing text about the benefits of being a member. The main form fields include 'Username' (alexanderbismark@gmail.com), 'Password (4 characters minimum)' (redacted), 'Password Confirm' (redacted), and 'Email' (alexanderbismark@gmail.com). Below the form is a terms of service agreement and a 'Sign Up' button.

**Sign up for an account**  
Signing up for an account is free and easy. Fill out the form below to get started. JavaScript is required to continue.

Use this option for  
create an account

of being a member

the movies and TV shows you have watched

- Keep track of your favourite movies and TV shows and get recommendations from them
- Build and maintain a personal watchlist
- Build custom mixed lists (movies and TV)
- Take part in movie and TV discussions
- Contribute to, and improve the information in our database

Username  
alexanderbismark@gmail.com

Password (4 characters minimum)  
\*\*\*\*\*

Password Confirm  
\*\*\*\*\*

Email  
alexanderbismark@gmail.com

By clicking the "Sign up" button below, I certify that I have read and agree to the TMDb terms of use and privacy policy.

**Sign Up** [Cancel](#)

You need verify your account on your email.

The screenshot shows the TMDb login page with a red header bar stating 'Account verification required'. It includes a message about verifying the email address and a list of steps. Below the header is a login form with 'Username' (alexanderbismark@gmail.com) and 'Password' (redacted) fields, along with 'Login' and 'Reset password' buttons.

**Login to your account**  
In order to use the editing and rating capabilities of TMDb, as well as get personal recommendations you will need to login to your account. If you do not have an account, registering for an account is free and simple. [Click here](#) to get started.

If you signed up but didn't get your verification email, [click here](#) to have it resent.

**Account verification required**

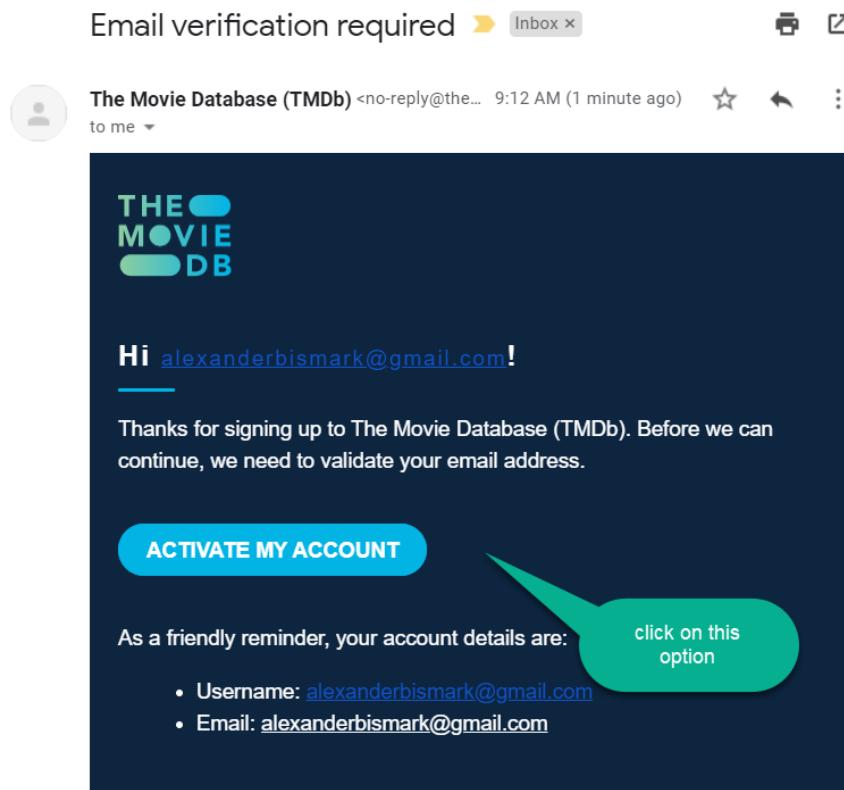
- Your email address hasn't been verified. Please click the verification link in the email that was sent to the address you signed up with. (Don't forget to check your spam folder.) You may [request the email be resent](#) if you are unable to locate your activation email.

Username  
alexanderbismark@gmail.com

Password  
\*\*\*\*\*

**Login** [Reset password](#)

Review your email and activate on email



Next

The image shows a screenshot of the TMDB website. At the top, there is a navigation bar with links for "Movies", "TV Shows", "People", "More", and language selection "EN". The main content area shows a user profile for "alexanderbismark@gmail.com". The profile picture is a portrait of a man in a suit. Below the picture are two circular icons: one for "Average Movie Score" (0%) and one for "Average TV Score" (0%). The "Watchlist" tab is highlighted in pink. A green speech bubble points to the "click on settings" link in the dropdown menu of the "Watchlist" tab. The dropdown menu also includes options like "Overview", "Discussions", "Lists", "Ratings", and "Edit Profile". At the bottom of the page, there is a footer with links for "THE MOVIE DB", "THE BASICS", "GET INVOLVED", "COMMUNITY", and "LEGAL". The "THE BASICS" section includes links for "About TMDb", "Contact Us", "Support Foru...", "API", and "System Status". The "GET INVOLVED" section includes "Contribution Bible", "3rd Party Applicati...", "Add New Movie", and "Add New TV Show". The "COMMUNITY" section includes "Guidelines", "Discussions", "Leaderboard", and "Twitter". The "LEGAL" section includes "Terms of Use", "API Terms of ..", and "Privacy Policy". A "Hi" greeting and the user's email address "alexanderbismark@gmail.com" are displayed at the bottom left.

Next

The screenshot shows the TMDB API Settings page. On the left, a sidebar titled 'Settings' lists various options: Edit Profile, Account Settings, Streaming Services, Notification Settings, Blocked Users, Import List, Sharing Settings, Social Settings, Connected Apps, and API. The 'API' link is highlighted with a green arrow pointing to a red 'CLICK HERE' button in the main content area. The main content area has tabs for API, Details, and Create. It contains text about TMDb's API service and links to documentation and support forums. A 'Request an API Key' section includes a button to click here.

Next

The screenshot shows the TMDB API Settings page with the user profile 'alexanderbismark@gmail.com' at the top. The sidebar shows the 'Settings' menu. In the main content area, a question asks 'What type of API key do you wish to register?'. Two options are shown: 'Developer' (selected) and 'Professional'. The 'Developer' option is described as follows: 'You are an individual', 'Your project is still in development', 'Your project is non profit', and 'Your project is ad supported'. A green callout bubble points to the 'use this option' text.

## Read and accept Use Terms and complete information

THE STATE OF CALIFORNIA WITHOUT REGARD TO ITS CONFLICT OF LAW PROVISIONS, YOU AND TMDB AGREE TO SUBMIT TO THE PERSONAL JURISDICTION OF THE COURTS LOCATED WITHIN THE COUNTY OF SAN MATEO, CALIFORNIA.

4. **No Waiver of Rights by TMDB.** TMDb's failure to exercise or enforce any right or provision of the Terms of Use shall not constitute a waiver of such right or provision.

5. **No Transfer.** The rights and obligations of these Terms of Use is personal to you and may not be transferred by you, either voluntarily or by operation of law.

6. **Notice.** Any notice to be sent to you under these Terms of Use may be sent via email, post, or any other reasonable means, at the contact information provided by you to TMDb from time to time. It is your obligation to insure that this information is current.

**Miscellaneous.** The section headings and subheadings contained in this agreement are included for convenience only, and shall not limit or otherwise affect the terms of the Terms of Use. Any construction or interpretation to be made of the Terms of Use shall not be construed against the drafter. The Terms of Use constitute the entire agreement between TMDb and you with respect to the subject matter hereof.

This Agreement was last updated on: July 28, 2014.

[Cancel](#) [Accept](#)



A screenshot of the TMDB API settings page. At the top, there is a navigation bar with 'TMDB' and links for 'Movies', 'TV Shows', 'People', 'More', a language switcher ('EN'), a notification bell, and a search icon. Below the navigation is a user profile section with a placeholder email 'alexanderbismark@gmail.com'. The main content area has a pink header 'Settings' and a sub-header 'API Details Create'. On the left, a sidebar lists 'Edit Profile', 'Account Settings', 'Streaming Services', 'Notification Settings', 'Blocked Users', 'Import List', 'Sharing Settings', 'Social Settings', and 'Connected Apps'. The main form fields include 'Type of Use' set to 'Mobile Application', 'Application Name' set to 'croc-movie', 'Application URL' set to 'https://github.com/alexanderbismark007/croc-movie', and an 'Application Summary' field containing the text 'Application for access to information'. At the bottom, there are fields for 'First Name' (set to 'Alexander') and 'Last Name' (set to 'Benitez').

Great work, its all

The screenshot shows the TMDB API Settings page. On the left, there's a sidebar with options like Edit Profile, Account Settings, Streaming Services, Notification Settings, Blocked Users, Import List, Sharing Settings, Social Settings, Connected Apps, API (which is selected), and Delete Account. The main content area has sections for API Details, Documentation, Support, and API Details. Under API Details, it says "If you'd like to edit the details of your app, click here." Below that is a "Social Directory" section with a note about adding the app to the directory. The "API Key (v3 auth)" section contains the key value "5ba8cc2e532ca3d5644bb39b0fe0c960". At the bottom, there's an "Example API Request" section with a code snippet: `https://api.themoviedb.org/3/movie/550?api_key=5ba8cc2e532ca3d5644bb39b0fe0c960`. Below that is an "API Read Access Token (v4 auth)" section with a long token value.

### API Key (v3 auth):

5ba8cc2e532ca3d5644bb39b0fe0c960

### Example API Request

[https://api.themoviedb.org/3/movie/550?api\\_key=5ba8cc2e532ca3d5644bb39b0fe0c960](https://api.themoviedb.org/3/movie/550?api_key=5ba8cc2e532ca3d5644bb39b0fe0c960)

### API Read Access Token (v4 auth)

eyJhbGciOiJIUzI1NiJ9eyJhdWQiOiIYmE4Y2MyZTUzMmNhM2Q1NjQ0YmlzOWIwZmUwYzk2MCIsInN1YiI6IjVlZjc0NWQ2MzlhMWE2MDAzNjjmMDNmZilsInNjb3BlcyI6WyJhcGlfcmVhZCJdLCJ2ZXJzaW9uljoxfQ.ktl4gmWCZhjhgGYryotGdV-FSq3MJdKn6kpKlrwxqpQ

### Access to DEVELOPER Environment

<https://developers.themoviedb.org/3/getting-started/introduction>

The screenshot shows the The Movie Database API documentation page. The top navigation bar includes links for THE MOVIE DB, THE BASICS, GET INVOLVED, COMMUNITY, and LEGAL. The main content area has a sidebar with sections like GETTING STARTED, ACCOUNT, AUTHENTICATION, CERTIFICATIONS, CHANGES, COLLECTIONS, COMPANIES, CONFIGURATION, and CREDITS. The AUTHENTICATION section is expanded, showing a sub-section titled "How do I generate a session id?". It provides instructions for generating a session ID, mentioning steps 1, 2, and 3. Step 1 is "Create a new request token". Step 2 is "Get the user to authorize the request token". Step 3 is "Create a new session id with the authorized request token". Below these steps, there's a "Step 1: Create a request token" section with a note about requesting a temporary token for permission. A URL is provided: `https://www.themoviedb.org/authenticate/(REQUEST_TOKEN)`.

## Verifying API Key and Uri via Postman

The screenshot shows the TMDB API documentation on the left and the Postman application interface on the right.

**TMDB API Documentation:**

- The sidebar lists various categories: ACCOUNT, AUTHENTICATION, CERTIFICATIONS, CHANGES, COLLECTIONS, COMPANIES, CONFIGURATION, CREDITS, DISCOVER, FIND, GENRES, GUEST SESSIONS, KEYWORDS, and LISTS.
- The AUTHENTICATION section details the "Create Guest Session" endpoint (`GET /authentication/guest_session/new`).
- The "Try it out" button is visible.

**Postman Interface:**

- The URL bar shows the endpoint: `https://api.themoviedb.org/3/authentication/guest_session/new?api_key=<>`.
- The "Authorization" tab is selected, showing "No Auth".
- The "Body" tab displays the JSON response:

```
1 ~ {  
2     "success": true,  
3     "guest_session_id": "b505b83a877a8ad3bcd14f7ee8925d77",  
4     "expires_at": "2020-06-28 13:42:21 UTC"  
5 }
```
- The status bar indicates "Status: 200 OK" and "Time: 462 ms".

## Get the last 10 popular movies

See API description for send parameters

The screenshot shows the 'Discover' section of the API documentation. On the left, there's a sidebar with various categories like 'GETTING STARTED', 'ACCOUNT', 'AUTHENTICATION', etc., and a 'DISCOVER' section highlighted with a green box. Under 'DISCOVER', there are two options: 'GET Movie Discover' and 'GET TV Discover'. The main content area is titled 'Movie Discover' and describes the 'GET /discover/movie' endpoint. It explains that it retrieves movies by different types of data like average rating, number of votes, genres and certifications. It also supports sort options and release date filtering. Below the description is a 'Definition' button and a 'Try it out' button.

### Example

- **LIST**

[https://api.themoviedb.org/3/discover/movie?api\\_key=5ba8cc2e532ca3d5644bb39b0fe0c960&language=en-US&sort\\_by=popularity.desc&include\\_adult=false&include\\_video=false&page=10](https://api.themoviedb.org/3/discover/movie?api_key=5ba8cc2e532ca3d5644bb39b0fe0c960&language=en-US&sort_by=popularity.desc&include_adult=false&include_video=false&page=10)

- **IMAGE**

<https://image.tmdb.org/t/p/w300/hjQp148VjWF4KjzhsD90OCMr11h.jpg>

- **DETAILS**

[https://api.themoviedb.org/3/movie/545609?api\\_key=5ba8cc2e532ca3d5644bb39b0fe0c960&language=en-US](https://api.themoviedb.org/3/movie/545609?api_key=5ba8cc2e532ca3d5644bb39b0fe0c960&language=en-US)

- **CREDITS**

[https://api.themoviedb.org/3/movie/545609/credits?api\\_key=5ba8cc2e532ca3d5644bb39b0fe0c960&language=en-US](https://api.themoviedb.org/3/movie/545609/credits?api_key=5ba8cc2e532ca3d5644bb39b0fe0c960&language=en-US)

The screenshot shows a Postman workspace with a single collection named 'Discover'. A GET request is made to 'https://api.themoviedb.org/3/discover/movie?api\_key=5ba8cc2e532ca3d5644bb39b0fe0c960&language=en-US&sort\_by=popularity.desc&include\_adult=false&include\_video=false&page=10'. The response status is 200 OK and the time taken is 421 ms. The response body is displayed in JSON format, showing the first few results of the movie list. A green arrow points to the 'total\_results' field in the JSON response, which is highlighted with a red box.

```
1< {
2   "page": 10,
3   "total_results": 10000,
4   "total_pages": 500,
5   "results": [
6     {
7       "popularity": 0.75,
8       "id": 719261,
9       "video": false,
10      "adult": 0,
11      "vote_average": 6,
12      "title": "Let Meotta Here",
13      "release_date": "2018-01-01",
14      "original_language": "en",
15      "original_title": "Let Meotta Here",
16      "genre_ids": [],
17      "backdrop_path": "/2v09uOpV5010UMWDX82eo9hx2.jpg",
18      "adult": false,
19      "overview": "A woman who has been estranged from her mother for years is forced to confront her past when her mother's sudden death leaves her with her mother's belongings, including a diary that reveals a dark secret."}
```

IMPORTANT: This API allows to obtain the information in English and Spanish;

## CREATING APP ON IONIC

For conventional use I prefers write CMD: when I use a Console independent if this Linux or Windows, Then I write the command. (on Linux use sudo previous any command)

CMD: ionic start croc-movie tabs

```
Microsoft Windows [Versión 10.0.18362.900]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

D:\ionic\croc-films>cd ..

D:\ionic>ionic start croc-movie tabs

Pick a framework!

Please select the JavaScript framework to use.
--type option.

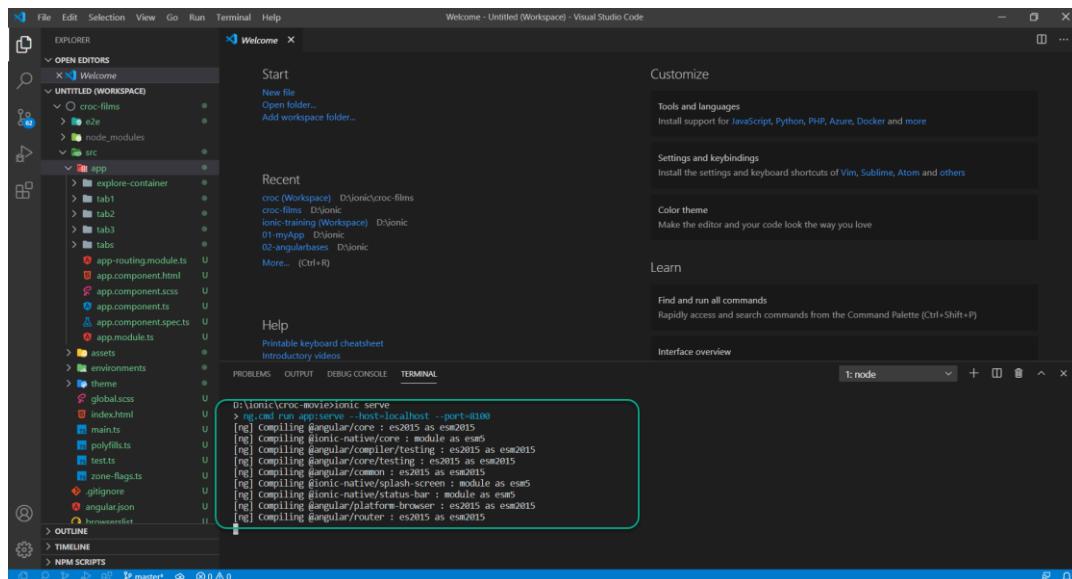
? Framework: (Use arrow keys)
  > Angular | https://angular.io
  React   | https://reactjs.org
```

Select your favorite Framework, I will use Angular for this DEMO

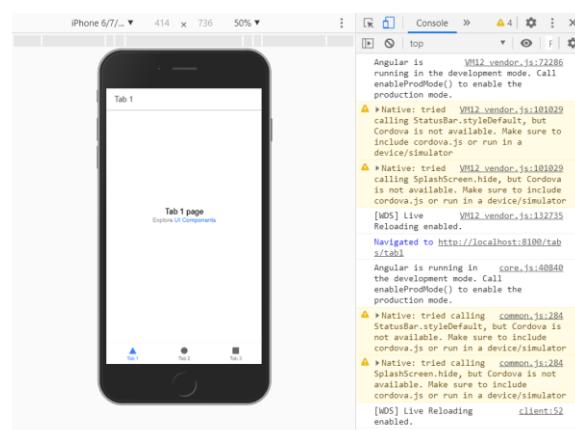
This prompt next time, supply a value for the --type option.

STARTING PROJECT (For my comfort, I enable auto save fie on menu File 😊)

CMD: ionic serve



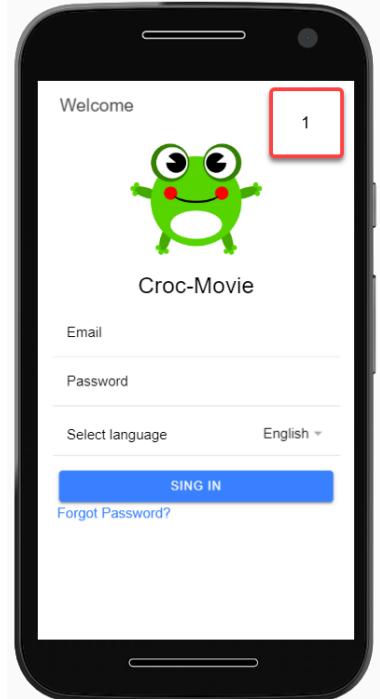
ENVIRONMENT ON CHROME (press F12 to get details and developer tools)



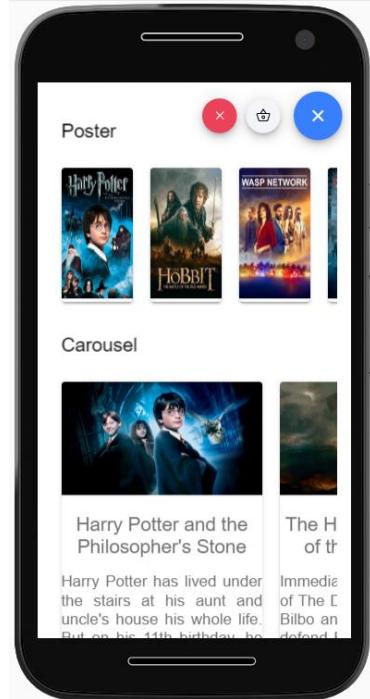
## DETALIS OF VERSIONS (package.json)

```
{  
  "name": "croc-films",  
  "version": "0.0.1",  
  "author": "Alexander Benitez Antelo",  
  "homepage": " https://github.com/alexanderbismark007/croc-movie",  
  "scripts": {  
    "ng": "ng",  
    "start": "ng serve",  
    "build": "ng build",  
    "test": "ng test",  
    "lint": "ng lint",  
    "e2e": "ng e2e"  
  },  
  "private": true,  
  "dependencies": {  
    "@angular/common": "~9.1.6",  
    "@angular/core": "~9.1.6",  
    "@angular/forms": "~9.1.6",  
    "@angular/platform-browser": "~9.1.6",  
    "@angular/platform-browser-dynamic": "~9.1.6",  
    "@angular/router": "~9.1.6",  
    "@ionic-native/core": "^5.0.7",  
    "@ionic-native/splash-screen": "^5.0.0",  
    "@ionic-native/status-bar": "^5.0.0",  
    "@ionic/angular": "^5.0.0",  
    "rxjs": "~6.5.1",  
    "tslib": "1.10.0",  
    "zone.js": "~0.10.2"  
  },  
  "devDependencies": {  
    "@angular-devkit/build-angular": "~0.901.5",  
    "@angular/cli": "~9.1.5",  
    "@angular/compiler": "~9.1.6",  
    "@angular/compiler-cli": "~9.1.6",  
    "@angular/language-service": "~9.1.6",  
    "@ionic/angular-toolkit": "2.1.1",  
    "@types/node": "12.11.1",  
    "@types/jasmine": "~3.5.0",  
    "@types/jasminewd2": "~2.0.3",  
    "codelyzer": "5.1.2",  
    "jasmine-core": "~3.5.0",  
    "jasmine-spec-reporter": "4.2.1",  
    "karma": "~5.0.0",  
    "karma-chrome-launcher": "~3.1.0",  
    "karma-coverage-istanbul-reporter": "~2.1.0",  
    "karma-jasmine": "~3.0.1",  
    "karma-jasmine-html-reporter": "1.4.2",  
    "protractor": "5.4.3",  
    "ts-node": "~8.3.0",  
    "tslint": "~6.1.0",  
    "typescript": "3.8.3"  
  },  
  "description": "CROC-MOVIE"  
}
```

**1- At least 2 pages: Master and Detail. (If you can add a mock login page will be great).**



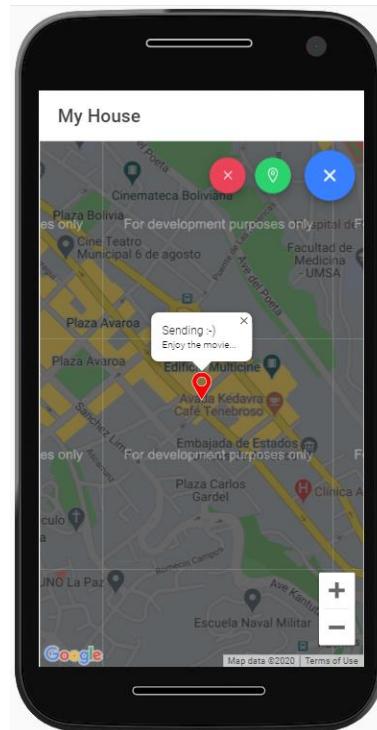
LOGIN



MASTER

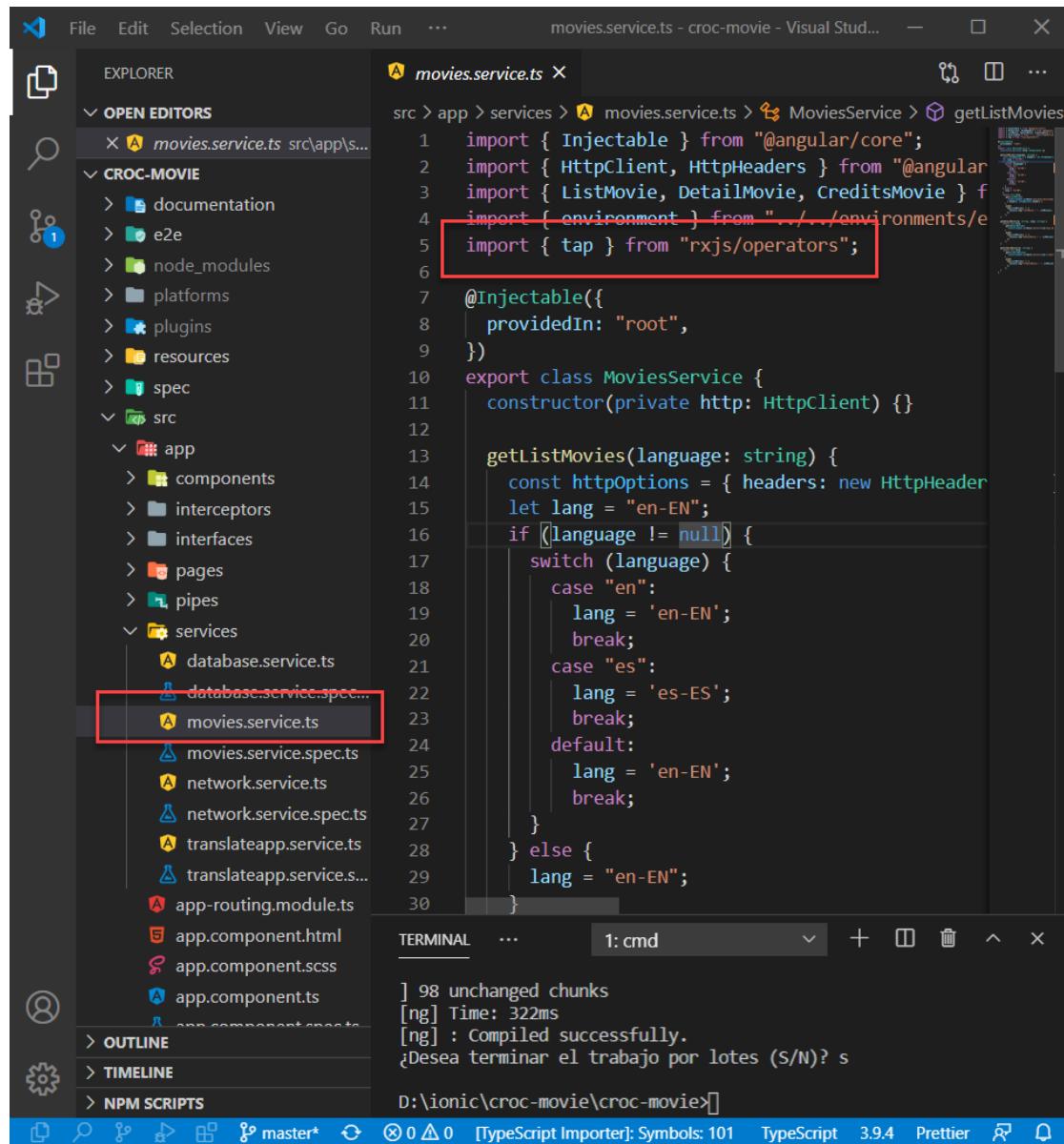


DETAIL



MAPS

## 2- Use RxJs to handle all the information from endpoints.



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "CROC-MOVIE". The file "movies.service.ts" is selected and highlighted with a red box.
- Editor:** Displays the code for "movies.service.ts". A specific line of code, `import { tap } from "rxjs/operators";`, is highlighted with a red box.
- Terminal:** Shows the command-line output of a build or compilation process, indicating 98 unchanged chunks, a total time of 322ms, and successful compilation.
- Bottom Status Bar:** Shows the current branch as "master\*", the number of changes as "0 △ 0", and the status bar message "[TypeScript Importer]: Symbols: 101 TypeScript 3.9.4 Prettier".

```
import { Injectable } from '@angular/core';
import { HttpClient, HttpHeaders } from '@angular/common/http';
import { ListMovie, DetailMovie, CreditsMovie } from '@angular/router';
import { environment } from '../../environments/environment';
import { tap } from 'rxjs/operators';

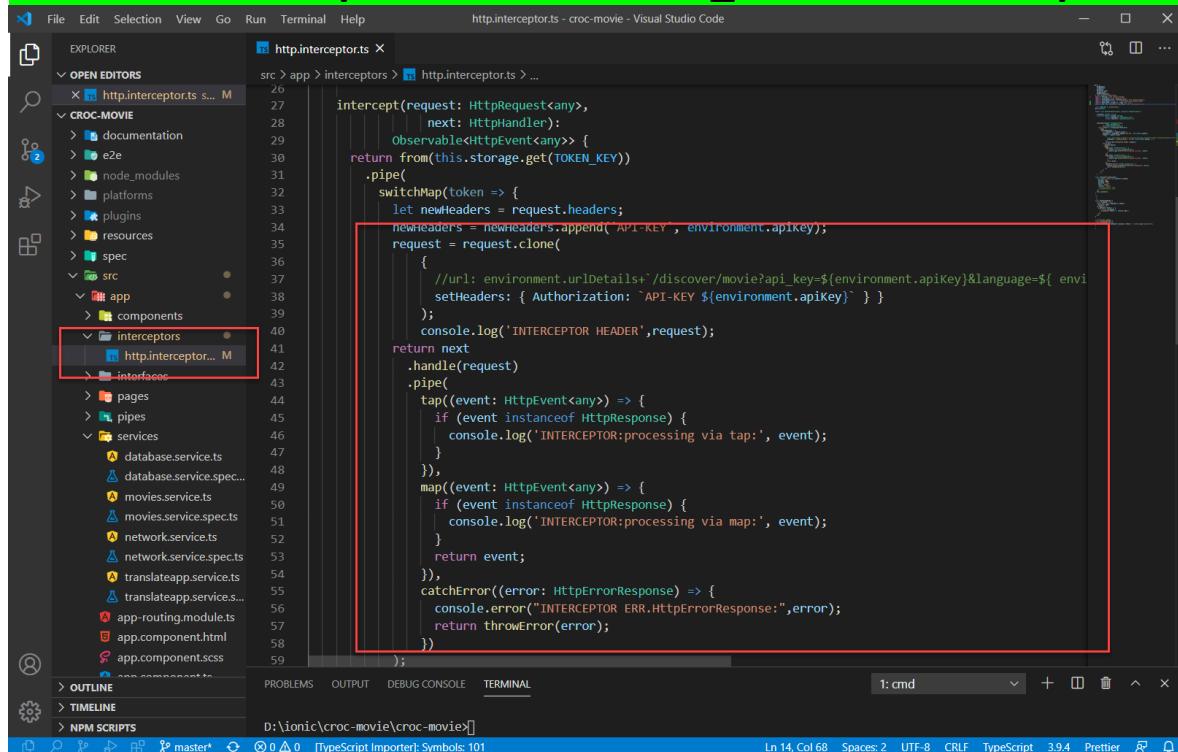
@Injectable({
  providedIn: 'root',
})
export class MoviesService {
  constructor(private http: HttpClient) {}

  getListMovies(language: string) {
    const httpOptions = { headers: new HttpHeaders({ 'Content-Type': 'application/json' }) };
    let lang = 'en-EN';
    if (!language != null) {
      switch (language) {
        case "en":
          lang = 'en-EN';
          break;
        case "es":
          lang = 'es-ES';
          break;
        default:
          lang = 'en-EN';
          break;
      }
    } else {
      lang = "en-EN";
    }
  }
}
```

The screenshot shows the Visual Studio Code interface with the following details:

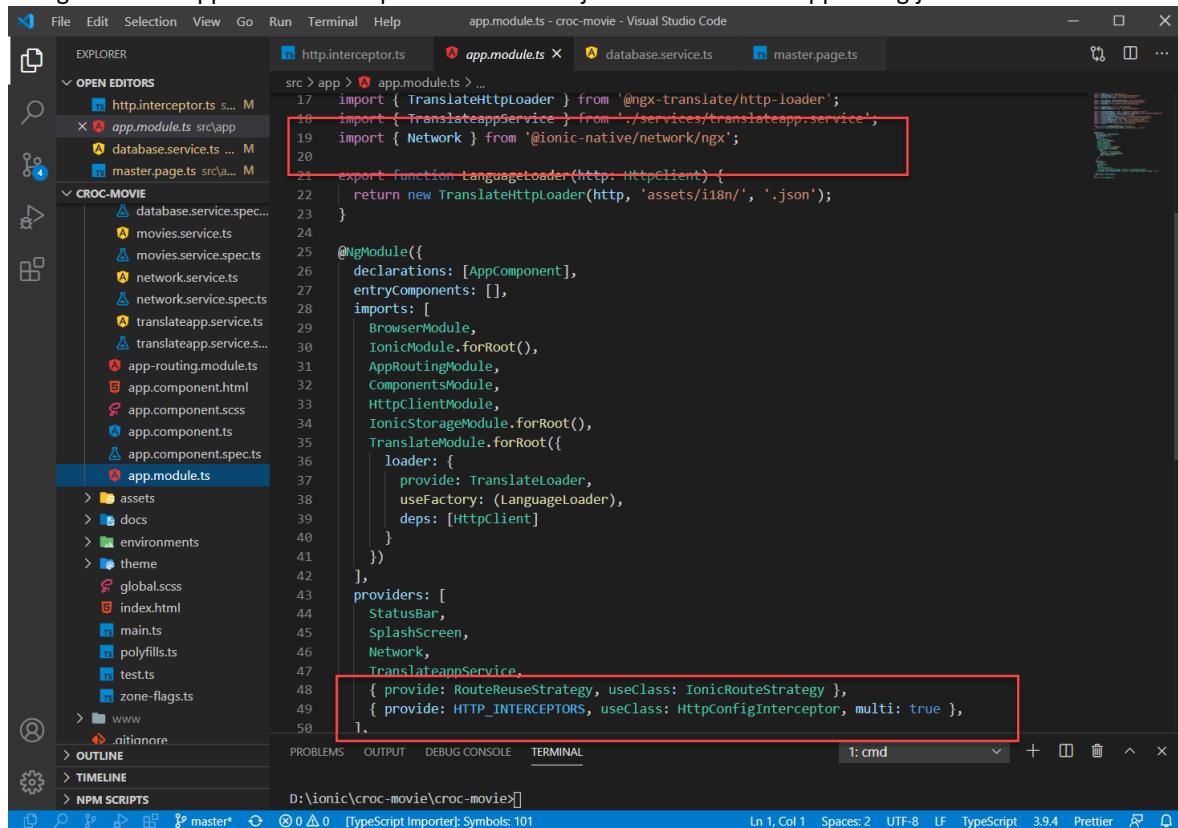
- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Editor Title:** movies.service.ts - croc-movie - Visual Studio...
- Explorer:** Shows the project structure under "CROC-MOVIE". The "src" folder contains "app" which has "components", "interceptors", "interfaces", "pages", "pipes", and "services". Inside "services", there are files: database.service.ts, database.service.spec.ts, movies.service.ts (selected), movies.service.spec.ts, network.service.ts, network.service.spec.ts, translateapp.service.ts, and translateapp.service.spec.ts.
- Code Editor:** The "movies.service.ts" file is open. Two sections of code are highlighted with red boxes:
  - The first section starts at line 49: `tap((lstMovies) => {` and ends at line 53: `});` It logs the detail movie list to the console.
  - The second section starts at line 61: `tap((lstMovies) => {` and ends at line 65: `});` It logs the credits movie list to the console.
- Terminal:** The terminal shows the output of a build command: "98 unchanged chunks", "Time: 322ms", and "Compiled successfully." It also shows the prompt "Desea terminar el trabajo por lotes (S/N)? s".
- Bottom Status Bar:** Shows the current branch as "master\*", file count as "0 △ 0", and build status as "TypeScript Importer: Symbols: 101 TypeScript 3.9.4 Prettier".

### 3- Use an interceptor to add the API\_KEY to all the requests.



```
src > app > interceptors > http.interceptor.ts > ...
26     intercept(request: HttpRequest<any>,
27             next: HttpHandler):
28         Observable<HttpEvent<any>> {
29             return from(this.storage.get(TOKEN_KEY))
30                 .pipe(
31                     switchMap(token => {
32                         let newHeaders = request.headers;
33                         newHeaders = newHeaders.append('API_KEY', environment.apiKey);
34                         request = request.clone(
35                             {
36                                 //url: environment.urlDetails+'/discover/movie?api_key=${environment.apiKey}&language=${ environment.language }
37                                 setHeaders: { Authorization: `API_KEY ${environment.apiKey}` }
38                             });
39                         console.log('INTERCEPTOR HEADER',request);
40                     });
41             return next
42                 .handle(request)
43                 .pipe(
44                     tap((event: HttpEvent<any>) => {
45                         if (event instanceof HttpResponse) {
46                             console.log('INTERCEPTOR:processing via tap:', event);
47                         }
48                     }),
49                     map((event: HttpEvent<any>) => {
50                         if (event instanceof HttpResponse) {
51                             console.log('INTERCEPTOR:processing via map:', event);
52                         }
53                         return event;
54                     }),
55                     catchError((error: HttpErrorResponse) => {
56                         console.error("INTERCEPTOR ERR.HttpErrorResponse:",error);
57                         return throwError(error);
58                     });
59                 );
60         );
61     );
62 }
```

Using RXJs for II application on Request an Receive inject this module on app.config.json



```
src > app > app.module.ts > ...
17     import { TranslateHttpLoader } from '@ngx-translate/http-loader';
18     import { TranslateAppService } from './services/translateapp.service';
19     import { Network } from '@ionic-native/network/ngx';
20
21     export function LanguageLoader(http: HttpClient) {
22         return new TranslateHttpLoader(http, 'assets/i18n/', '.json');
23     }
24
25     @NgModule({
26         declarations: [AppComponent],
27         entryComponents: [],
28         imports: [
29             BrowserModule,
30             IonicModule.forRoot(),
31             AppRoutingModule,
32             ComponentsModule,
33             HttpClientModule,
34             IonicStorageModule.forRoot(),
35             TranslateModule.forRoot({
36                 loader: {
37                     provide: TranslateLoader,
38                     useFactory: (LanguageLoader),
39                     deps: [HttpClient]
40                 }
41             })
42         ],
43         providers: [
44             StatusBar,
45             SplashScreen,
46             Network,
47             TranslateAppService,
48             { provide: RouteReuseStrategy, useClass: IonicRouteStrategy },
49             { provide: HTTP_INTERCEPTORS, useClass: HttpConfigInterceptor, multi: true },
50         ],
51     })
52 }
```

The screenshot shows a Moto G4 browser window displaying a mobile application. On the left, there's a smartphone icon representing the device. Below it, two sections are visible: "Poster" and "Carousel". The "Poster" section shows three movie posters: Harry Potter, Hobbit, and Star Wars. The "Carousel" section shows a larger image of Harry Potter and the Philosopher's Stone.

On the right side of the browser window, the developer tools are open. The "Network" tab is selected, showing a timeline with several requests. A table below the timeline lists the requests:

Name	Status	Type	Initiator	Size	Time	Waterfall
background.js	304	script	cordova.js[1312]	242 B	24 ms	
CameraPosition.js	304	script	cordova.js[1312]	243 B	23 ms	
Polyline.js	304	script	cordova.js[1312]	243 B	28 ms	
Polygon.js	304	script	cordova.js[1312]	243 B	30 ms	
Marker.js	304	script	cordova.js[1312]	243 B	28 ms	
HtmlInfoWindow.js	304	script	cordova.js[1312]	243 B	28 ms	
Circle.js	304	script	cordova.js[1312]	243 B	28 ms	
TileOverlay.js	304	script	cordova.js[1312]	242 B	29 ms	
GroundOverlay.js	304	script	cordova.js[1312]	243 B	30 ms	
encoding.js	304	script	cordova.js[1312]	242 B	27 ms	

At the bottom of the Network tab, it says: 173 requests | 448 kB transferred | 9.1 MB resources | Finish: 4.98 s | DOMContentLoaded: 1.14 s | Load: 2.04 s

Below the Network tab, the "Console" tab is also visible, showing some log entries. One entry is highlighted with a red box:

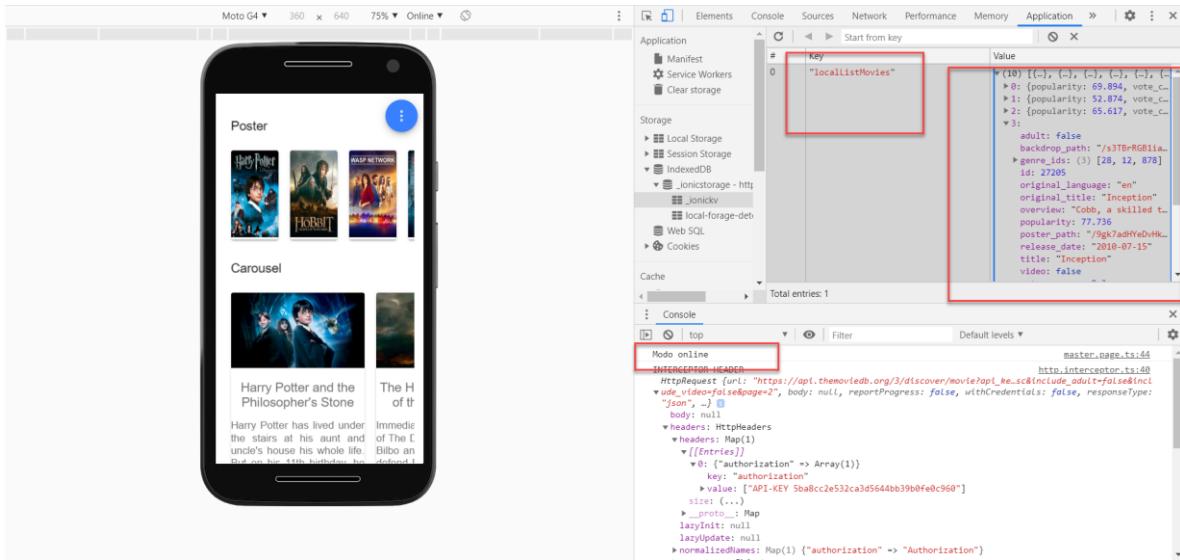
```
INTERATOR HEADER
HttpHeader [url: "https://api.themoviedb.org/3/discover/movie?api_key=...&include_adult=false&language=en-US&sort_by=popularity.desc&vote_average.gte=8.2", body: null, reportProgress: false, withCredentials: false, responseType: "json", ...]
  body: null
  headers: HttpHeaders
    +headers: HttpHeaders
      +entries: Map
        +0: {"authorization": Array(1)}
          key: "authorization"
          value: ["API-KEY 5ba8cc2e532ca3d5644bb39b0fe0c960"]
          size: {...}
        ...
      lazyInit: null
      lazyUpdate: null
    +normalizedNames: Map(1) {"authorization" => "Authorization"}
```

Show Headers and request on Debug mode

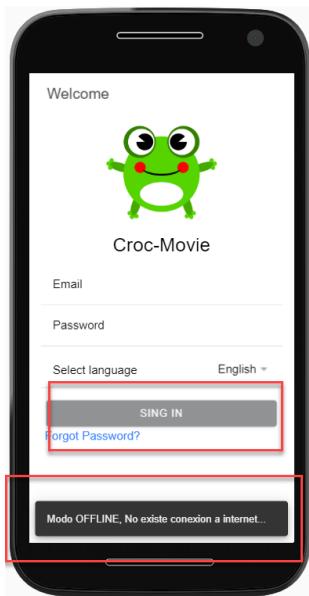
4- Save the results from the API call in localStorage/database in order to show the info immediately if the app doesn't have internet access.

I have created modules that detect if have connection to internet, after when it's online, save data in other cases load local data.

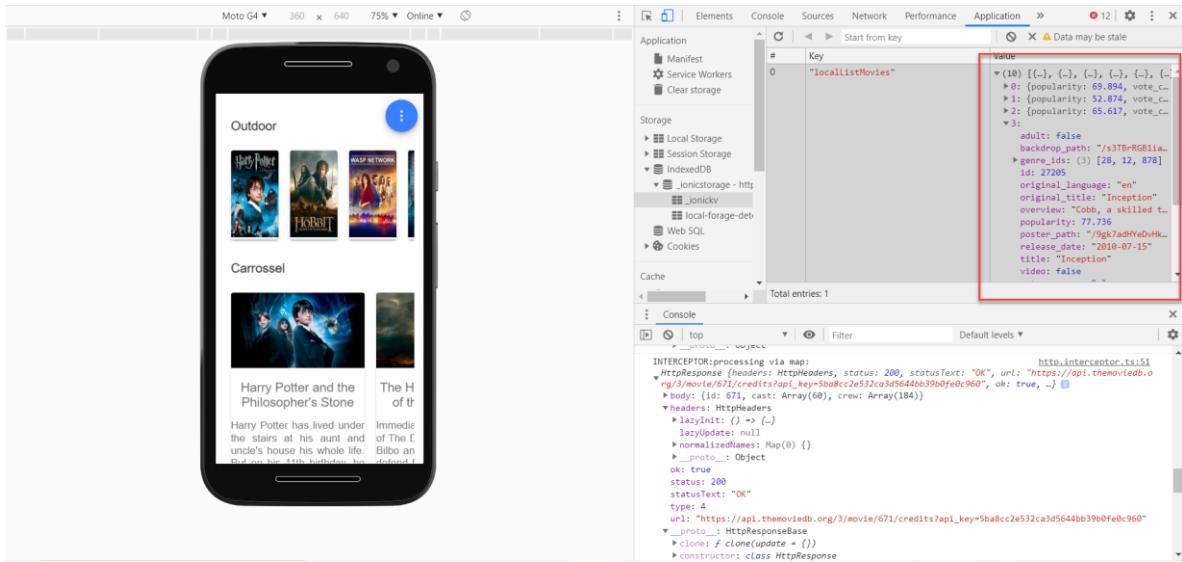
I have created a service that manage all request for local state



When APP is online, save all Data, the imagen shows this purpose.



When connection is OFFLINE shows a message and change colors of APP



## Access to OFFLINE mode and get Local Data

## 5- Use at least one External Library or Cordova Plugin. Ej. GPS, Internet connection viewer, get UUID from device, Permissions, etc.

```

File Edit Selection View Go Run Terminal Help
maps.page.ts - croc-movie - Visual Studio Code
src > app > pages > maps > maps.page.ts > ...
1 import { Component, OnInit } from '@angular/core';
2
3 import { GoogleMaps, GoogleMap, GoogleMapsEvent, Marker, GoogleMapsAnimation, MyLocation
4 import { Platform, LoadingController, ToastController, ModalController, NavController } f
5 import { TranslateappService } from '../../../../../services/translateapp.service';
6
7 @Component({
8   selector: "app-maps",
9   templateUrl: "./maps.page.html",
10  styleUrls: ["./maps.page.scss"],
11})
12 export class MapsPage implements OnInit {
13   map: GoogleMap;
14   loading: any;
15   selectedLanguage:string;
16
17 constructor( public loadingCtrl: LoadingController,
18   private navCtrl: NavController,
19   public toastCtrl: ToastController,
20   private platform: Platform,
21   private translateConfigService: TranslateappService ) {
22   this.selectedLanguage = this.translateConfigService.getLanguage();
23 }
24
25 async ngOnInit() {
26   await this.platform.ready();
27   await this.loadMap();
28 }
29
30 loadMap() {
31
32   this.map = GoogleMaps.create("map_canvas", {
33     camera: {
34       target: {
35         lat: 40.4128,
36         lon: -3.7025
37       },
38       zoom: 15
39     }
40   });
41
42   this.map.on(GoogleMapsEvent.MAP_READY).subscribe(() => {
43     this.loadingCtrl.dismiss();
44   });
45 }
46
47 ionViewDidEnter() {
48   this.map.setZoom(15);
49 }
50
51 ionViewWillLeave() {
52   this.map.setZoom(10);
53 }
54
55 ionViewDidLeave() {
56   this.map.setZoom(15);
57 }
58
59 ionViewWillEnter() {
60   this.map.setZoom(15);
61 }
62
63 ionViewDidEnter() {
64   this.map.setZoom(15);
65 }
66
67 ionViewWillLeave() {
68   this.map.setZoom(10);
69 }
70
71 ionViewDidLeave() {
72   this.map.setZoom(15);
73 }
74
75 ionViewWillEnter() {
76   this.map.setZoom(15);
77 }
78
79 ionViewDidEnter() {
80   this.map.setZoom(15);
81 }
82
83 ionViewWillLeave() {
84   this.map.setZoom(10);
85 }
86
87 ionViewDidLeave() {
88   this.map.setZoom(15);
89 }
90
91 ionViewWillEnter() {
92   this.map.setZoom(15);
93 }
94
95 ionViewDidEnter() {
96   this.map.setZoom(15);
97 }
98
99 ionViewWillLeave() {
100  this.map.setZoom(10);
101 }
102
103 ionViewDidLeave() {
104   this.map.setZoom(15);
105 }
106
107 ionViewWillEnter() {
108   this.map.setZoom(15);
109 }
110
111 ionViewDidEnter() {
112   this.map.setZoom(15);
113 }
114
115 ionViewWillLeave() {
116   this.map.setZoom(10);
117 }
118
119 ionViewDidLeave() {
120   this.map.setZoom(15);
121 }
122
123 ionViewWillEnter() {
124   this.map.setZoom(15);
125 }
126
127 ionViewDidEnter() {
128   this.map.setZoom(15);
129 }
130
131 ionViewWillLeave() {
132   this.map.setZoom(10);
133 }
134
135 ionViewDidLeave() {
136   this.map.setZoom(15);
137 }
138
139 ionViewWillEnter() {
140   this.map.setZoom(15);
141 }
142
143 ionViewDidEnter() {
144   this.map.setZoom(15);
145 }
146
147 ionViewWillLeave() {
148   this.map.setZoom(10);
149 }
150
151 ionViewDidLeave() {
152   this.map.setZoom(15);
153 }
154
155 ionViewWillEnter() {
156   this.map.setZoom(15);
157 }
158
159 ionViewDidEnter() {
160   this.map.setZoom(15);
161 }
162
163 ionViewWillLeave() {
164   this.map.setZoom(10);
165 }
166
167 ionViewDidLeave() {
168   this.map.setZoom(15);
169 }
170
171 ionViewWillEnter() {
172   this.map.setZoom(15);
173 }
174
175 ionViewDidEnter() {
176   this.map.setZoom(15);
177 }
178
179 ionViewWillLeave() {
180   this.map.setZoom(10);
181 }
182
183 ionViewDidLeave() {
184   this.map.setZoom(15);
185 }
186
187 ionViewWillEnter() {
188   this.map.setZoom(15);
189 }
190
191 ionViewDidEnter() {
192   this.map.setZoom(15);
193 }
194
195 ionViewWillLeave() {
196   this.map.setZoom(10);
197 }
198
199 ionViewDidLeave() {
200   this.map.setZoom(15);
201 }
202
203 ionViewWillEnter() {
204   this.map.setZoom(15);
205 }
206
207 ionViewDidEnter() {
208   this.map.setZoom(15);
209 }
210
211 ionViewWillLeave() {
212   this.map.setZoom(10);
213 }
214
215 ionViewDidLeave() {
216   this.map.setZoom(15);
217 }
218
219 ionViewWillEnter() {
220   this.map.setZoom(15);
221 }
222
223 ionViewDidEnter() {
224   this.map.setZoom(15);
225 }
226
227 ionViewWillLeave() {
228   this.map.setZoom(10);
229 }
230
231 ionViewDidLeave() {
232   this.map.setZoom(15);
233 }
234
235 ionViewWillEnter() {
236   this.map.setZoom(15);
237 }
238
239 ionViewDidEnter() {
240   this.map.setZoom(15);
241 }
242
243 ionViewWillLeave() {
244   this.map.setZoom(10);
245 }
246
247 ionViewDidLeave() {
248   this.map.setZoom(15);
249 }
250
251 ionViewWillEnter() {
252   this.map.setZoom(15);
253 }
254
255 ionViewDidEnter() {
256   this.map.setZoom(15);
257 }
258
259 ionViewWillLeave() {
260   this.map.setZoom(10);
261 }
262
263 ionViewDidLeave() {
264   this.map.setZoom(15);
265 }
266
267 ionViewWillEnter() {
268   this.map.setZoom(15);
269 }
270
271 ionViewDidEnter() {
272   this.map.setZoom(15);
273 }
274
275 ionViewWillLeave() {
276   this.map.setZoom(10);
277 }
278
279 ionViewDidLeave() {
280   this.map.setZoom(15);
281 }
282
283 ionViewWillEnter() {
284   this.map.setZoom(15);
285 }
286
287 ionViewDidEnter() {
288   this.map.setZoom(15);
289 }
290
291 ionViewWillLeave() {
292   this.map.setZoom(10);
293 }
294
295 ionViewDidLeave() {
296   this.map.setZoom(15);
297 }
298
299 ionViewWillEnter() {
300   this.map.setZoom(15);
301 }
302
303 ionViewDidEnter() {
304   this.map.setZoom(15);
305 }
306
307 ionViewWillLeave() {
308   this.map.setZoom(10);
309 }
310
311 ionViewDidLeave() {
312   this.map.setZoom(15);
313 }
314
315 ionViewWillEnter() {
316   this.map.setZoom(15);
317 }
318
319 ionViewDidEnter() {
320   this.map.setZoom(15);
321 }
322
323 ionViewWillLeave() {
324   this.map.setZoom(10);
325 }
326
327 ionViewDidLeave() {
328   this.map.setZoom(15);
329 }
330
331 ionViewWillEnter() {
332   this.map.setZoom(15);
333 }
334
335 ionViewDidEnter() {
336   this.map.setZoom(15);
337 }
338
339 ionViewWillLeave() {
340   this.map.setZoom(10);
341 }
342
343 ionViewDidLeave() {
344   this.map.setZoom(15);
345 }
346
347 ionViewWillEnter() {
348   this.map.setZoom(15);
349 }
350
351 ionViewDidEnter() {
352   this.map.setZoom(15);
353 }
354
355 ionViewWillLeave() {
356   this.map.setZoom(10);
357 }
358
359 ionViewDidLeave() {
360   this.map.setZoom(15);
361 }
362
363 ionViewWillEnter() {
364   this.map.setZoom(15);
365 }
366
367 ionViewDidEnter() {
368   this.map.setZoom(15);
369 }
370
371 ionViewWillLeave() {
372   this.map.setZoom(10);
373 }
374
375 ionViewDidLeave() {
376   this.map.setZoom(15);
377 }
378
379 ionViewWillEnter() {
380   this.map.setZoom(15);
381 }
382
383 ionViewDidEnter() {
384   this.map.setZoom(15);
385 }
386
387 ionViewWillLeave() {
388   this.map.setZoom(10);
389 }
390
391 ionViewDidLeave() {
392   this.map.setZoom(15);
393 }
394
395 ionViewWillEnter() {
396   this.map.setZoom(15);
397 }
398
399 ionViewDidEnter() {
400   this.map.setZoom(15);
401 }
402
403 ionViewWillLeave() {
404   this.map.setZoom(10);
405 }
406
407 ionViewDidLeave() {
408   this.map.setZoom(15);
409 }
410
411 ionViewWillEnter() {
412   this.map.setZoom(15);
413 }
414
415 ionViewDidEnter() {
416   this.map.setZoom(15);
417 }
418
419 ionViewWillLeave() {
420   this.map.setZoom(10);
421 }
422
423 ionViewDidLeave() {
424   this.map.setZoom(15);
425 }
426
427 ionViewWillEnter() {
428   this.map.setZoom(15);
429 }
430
431 ionViewDidEnter() {
432   this.map.setZoom(15);
433 }
434
435 ionViewWillLeave() {
436   this.map.setZoom(10);
437 }
438
439 ionViewDidLeave() {
440   this.map.setZoom(15);
441 }
442
443 ionViewWillEnter() {
444   this.map.setZoom(15);
445 }
446
447 ionViewDidEnter() {
448   this.map.setZoom(15);
449 }
450
451 ionViewWillLeave() {
452   this.map.setZoom(10);
453 }
454
455 ionViewDidLeave() {
456   this.map.setZoom(15);
457 }
458
459 ionViewWillEnter() {
460   this.map.setZoom(15);
461 }
462
463 ionViewDidEnter() {
464   this.map.setZoom(15);
465 }
466
467 ionViewWillLeave() {
468   this.map.setZoom(10);
469 }
470
471 ionViewDidLeave() {
472   this.map.setZoom(15);
473 }
474
475 ionViewWillEnter() {
476   this.map.setZoom(15);
477 }
478
479 ionViewDidEnter() {
480   this.map.setZoom(15);
481 }
482
483 ionViewWillLeave() {
484   this.map.setZoom(10);
485 }
486
487 ionViewDidLeave() {
488   this.map.setZoom(15);
489 }
490
491 ionViewWillEnter() {
492   this.map.setZoom(15);
493 }
494
495 ionViewDidEnter() {
496   this.map.setZoom(15);
497 }
498
499 ionViewWillLeave() {
500   this.map.setZoom(10);
501 }
502
503 ionViewDidLeave() {
504   this.map.setZoom(15);
505 }
506
507 ionViewWillEnter() {
508   this.map.setZoom(15);
509 }
510
511 ionViewDidEnter() {
512   this.map.setZoom(15);
513 }
514
515 ionViewWillLeave() {
516   this.map.setZoom(10);
517 }
518
519 ionViewDidLeave() {
520   this.map.setZoom(15);
521 }
522
523 ionViewWillEnter() {
524   this.map.setZoom(15);
525 }
526
527 ionViewDidEnter() {
528   this.map.setZoom(15);
529 }
530
531 ionViewWillLeave() {
532   this.map.setZoom(10);
533 }
534
535 ionViewDidLeave() {
536   this.map.setZoom(15);
537 }
538
539 ionViewWillEnter() {
540   this.map.setZoom(15);
541 }
542
543 ionViewDidEnter() {
544   this.map.setZoom(15);
545 }
546
547 ionViewWillLeave() {
548   this.map.setZoom(10);
549 }
550
551 ionViewDidLeave() {
552   this.map.setZoom(15);
553 }
554
555 ionViewWillEnter() {
556   this.map.setZoom(15);
557 }
558
559 ionViewDidEnter() {
560   this.map.setZoom(15);
561 }
562
563 ionViewWillLeave() {
564   this.map.setZoom(10);
565 }
566
567 ionViewDidLeave() {
568   this.map.setZoom(15);
569 }
570
571 ionViewWillEnter() {
572   this.map.setZoom(15);
573 }
574
575 ionViewDidEnter() {
576   this.map.setZoom(15);
577 }
578
579 ionViewWillLeave() {
580   this.map.setZoom(10);
581 }
582
583 ionViewDidLeave() {
584   this.map.setZoom(15);
585 }
586
587 ionViewWillEnter() {
588   this.map.setZoom(15);
589 }
590
591 ionViewDidEnter() {
592   this.map.setZoom(15);
593 }
594
595 ionViewWillLeave() {
596   this.map.setZoom(10);
597 }
598
599 ionViewDidLeave() {
600   this.map.setZoom(15);
601 }
602
603 ionViewWillEnter() {
604   this.map.setZoom(15);
605 }
606
607 ionViewDidEnter() {
608   this.map.setZoom(15);
609 }
610
611 ionViewWillLeave() {
612   this.map.setZoom(10);
613 }
614
615 ionViewDidLeave() {
616   this.map.setZoom(15);
617 }
618
619 ionViewWillEnter() {
620   this.map.setZoom(15);
621 }
622
623 ionViewDidEnter() {
624   this.map.setZoom(15);
625 }
626
627 ionViewWillLeave() {
628   this.map.setZoom(10);
629 }
630
631 ionViewDidLeave() {
632   this.map.setZoom(15);
633 }
634
635 ionViewWillEnter() {
636   this.map.setZoom(15);
637 }
638
639 ionViewDidEnter() {
640   this.map.setZoom(15);
641 }
642
643 ionViewWillLeave() {
644   this.map.setZoom(10);
645 }
646
647 ionViewDidLeave() {
648   this.map.setZoom(15);
649 }
650
651 ionViewWillEnter() {
652   this.map.setZoom(15);
653 }
654
655 ionViewDidEnter() {
656   this.map.setZoom(15);
657 }
658
659 ionViewWillLeave() {
660   this.map.setZoom(10);
661 }
662
663 ionViewDidLeave() {
664   this.map.setZoom(15);
665 }
666
667 ionViewWillEnter() {
668   this.map.setZoom(15);
669 }
670
671 ionViewDidEnter() {
672   this.map.setZoom(15);
673 }
674
675 ionViewWillLeave() {
676   this.map.setZoom(10);
677 }
678
679 ionViewDidLeave() {
680   this.map.setZoom(15);
681 }
682
683 ionViewWillEnter() {
684   this.map.setZoom(15);
685 }
686
687 ionViewDidEnter() {
688   this.map.setZoom(15);
689 }
690
691 ionViewWillLeave() {
692   this.map.setZoom(10);
693 }
694
695 ionViewDidLeave() {
696   this.map.setZoom(15);
697 }
698
699 ionViewWillEnter() {
700   this.map.setZoom(15);
701 }
702
703 ionViewDidEnter() {
704   this.map.setZoom(15);
705 }
706
707 ionViewWillLeave() {
708   this.map.setZoom(10);
709 }
710
711 ionViewDidLeave() {
712   this.map.setZoom(15);
713 }
714
715 ionViewWillEnter() {
716   this.map.setZoom(15);
717 }
718
719 ionViewDidEnter() {
720   this.map.setZoom(15);
721 }
722
723 ionViewWillLeave() {
724   this.map.setZoom(10);
725 }
726
727 ionViewDidLeave() {
728   this.map.setZoom(15);
729 }
730
731 ionViewWillEnter() {
732   this.map.setZoom(15);
733 }
734
735 ionViewDidEnter() {
736   this.map.setZoom(15);
737 }
738
739 ionViewWillLeave() {
740   this.map.setZoom(10);
741 }
742
743 ionViewDidLeave() {
744   this.map.setZoom(15);
745 }
746
747 ionViewWillEnter() {
748   this.map.setZoom(15);
749 }
750
751 ionViewDidEnter() {
752   this.map.setZoom(15);
753 }
754
755 ionViewWillLeave() {
756   this.map.setZoom(10);
757 }
758
759 ionViewDidLeave() {
760   this.map.setZoom(15);
761 }
762
763 ionViewWillEnter() {
764   this.map.setZoom(15);
765 }
766
767 ionViewDidEnter() {
768   this.map.setZoom(15);
769 }
770
771 ionViewWillLeave() {
772   this.map.setZoom(10);
773 }
774
775 ionViewDidLeave() {
776   this.map.setZoom(15);
777 }
778
779 ionViewWillEnter() {
780   this.map.setZoom(15);
781 }
782
783 ionViewDidEnter() {
784   this.map.setZoom(15);
785 }
786
787 ionViewWillLeave() {
788   this.map.setZoom(10);
789 }
790
791 ionViewDidLeave() {
792   this.map.setZoom(15);
793 }
794
795 ionViewWillEnter() {
796   this.map.setZoom(15);
797 }
798
799 ionViewDidEnter() {
800   this.map.setZoom(15);
801 }
802
803 ionViewWillLeave() {
804   this.map.setZoom(10);
805 }
806
807 ionViewDidLeave() {
808   this.map.setZoom(15);
809 }
810
811 ionViewWillEnter() {
812   this.map.setZoom(15);
813 }
814
815 ionViewDidEnter() {
816   this.map.setZoom(15);
817 }
818
819 ionViewWillLeave() {
820   this.map.setZoom(10);
821 }
822
823 ionViewDidLeave() {
824   this.map.setZoom(15);
825 }
826
827 ionViewWillEnter() {
828   this.map.setZoom(15);
829 }
830
831 ionViewDidEnter() {
832   this.map.setZoom(15);
833 }
834
835 ionViewWillLeave() {
836   this.map.setZoom(10);
837 }
838
839 ionViewDidLeave() {
840   this.map.setZoom(15);
841 }
842
843 ionViewWillEnter() {
844   this.map.setZoom(15);
845 }
846
847 ionViewDidEnter() {
848   this.map.setZoom(15);
849 }
850
851 ionViewWillLeave() {
852   this.map.setZoom(10);
853 }
854
855 ionViewDidLeave() {
856   this.map.setZoom(15);
857 }
858
859 ionViewWillEnter() {
860   this.map.setZoom(15);
861 }
862
863 ionViewDidEnter() {
864   this.map.setZoom(15);
865 }
866
867 ionViewWillLeave() {
868   this.map.setZoom(10);
869 }
870
871 ionViewDidLeave() {
872   this.map.setZoom(15);
873 }
874
875 ionViewWillEnter() {
876   this.map.setZoom(15);
877 }
878
879 ionViewDidEnter() {
880   this.map.setZoom(15);
881 }
882
883 ionViewWillLeave() {
884   this.map.setZoom(10);
885 }
886
887 ionViewDidLeave() {
888   this.map.setZoom(15);
889 }
890
891 ionViewWillEnter() {
892   this.map.setZoom(15);
893 }
894
895 ionViewDidEnter() {
896   this.map.setZoom(15);
897 }
898
899 ionViewWillLeave() {
900   this.map.setZoom(10);
901 }
902
903 ionViewDidLeave() {
904   this.map.setZoom(15);
905 }
906
907 ionViewWillEnter() {
908   this.map.setZoom(15);
909 }
910
911 ionViewDidEnter() {
912   this.map.setZoom(15);
913 }
914
915 ionViewWillLeave() {
916   this.map.setZoom(10);
917 }
918
919 ionViewDidLeave() {
920   this.map.setZoom(15);
921 }
922
923 ionViewWillEnter() {
924   this.map.setZoom(15);
925 }
926
927 ionViewDidEnter() {
928   this.map.setZoom(15);
929 }
930
931 ionViewWillLeave() {
932   this.map.setZoom(10);
933 }
934
935 ionViewDidLeave() {
936   this.map.setZoom(15);
937 }
938
939 ionViewWillEnter() {
940   this.map.setZoom(15);
941 }
942
943 ionViewDidEnter() {
944   this.map.setZoom(15);
945 }
946
947 ionViewWillLeave() {
948   this.map.setZoom(10);
949 }
950
951 ionViewDidLeave() {
952   this.map.setZoom(15);
953 }
954
955 ionViewWillEnter() {
956   this.map.setZoom(15);
957 }
958
959 ionViewDidEnter() {
960   this.map.setZoom(15);
961 }
962
963 ionViewWillLeave() {
964   this.map.setZoom(10);
965 }
966
967 ionViewDidLeave() {
968   this.map.setZoom(15);
969 }
970
971 ionViewWillEnter() {
972   this.map.setZoom(15);
973 }
974
975 ionViewDidEnter() {
976   this.map.setZoom(15);
977 }
978
979 ionViewWillLeave() {
980   this.map.setZoom(10);
981 }
982
983 ionViewDidLeave() {
984   this.map.setZoom(15);
985 }
986
987 ionViewWillEnter() {
988   this.map.setZoom(15);
989 }
990
991 ionViewDidEnter() {
992   this.map.setZoom(15);
993 }
994
995 ionViewWillLeave() {
996   this.map.setZoom(10);
997 }
998
999 ionViewDidLeave() {
1000  this.map.setZoom(15);
1001 }
```

I have created a Page that use Components with libraries

- npm install @ionic-native/core@beta @ionic-native/google-maps@beta
- ionic cordova plugin add [https://github.com/mapsplugin/cordova-plugin-googlemaps#multiple\\_maps](https://github.com/mapsplugin/cordova-plugin-googlemaps#multiple_maps)

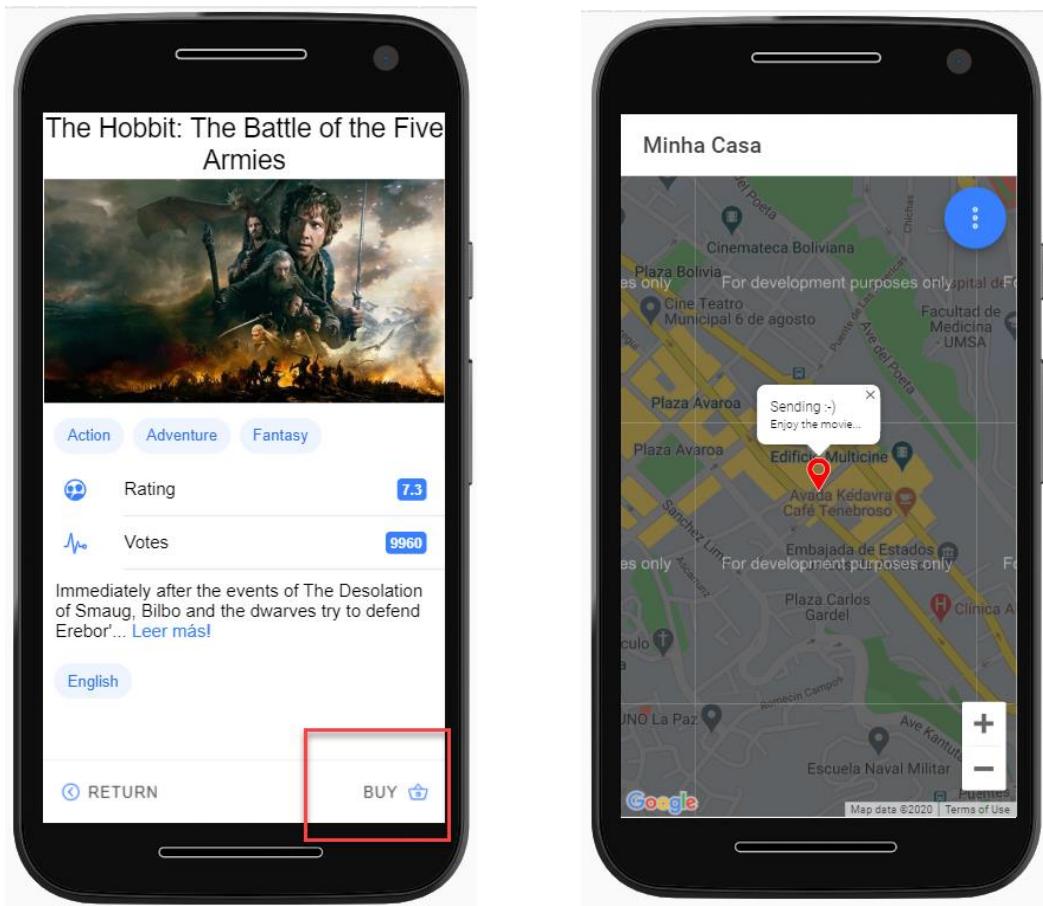
ADD this line on config.XML with API KEY of DEVELOPER GOOGLE

```
<preference name="GOOGLE_MAPS_ANDROID_API_KEY" value="NUESTRO_API_KEY" />
```

Execute this command (*ionic serve* not is compatible for this option)

- ionic cordova run browser

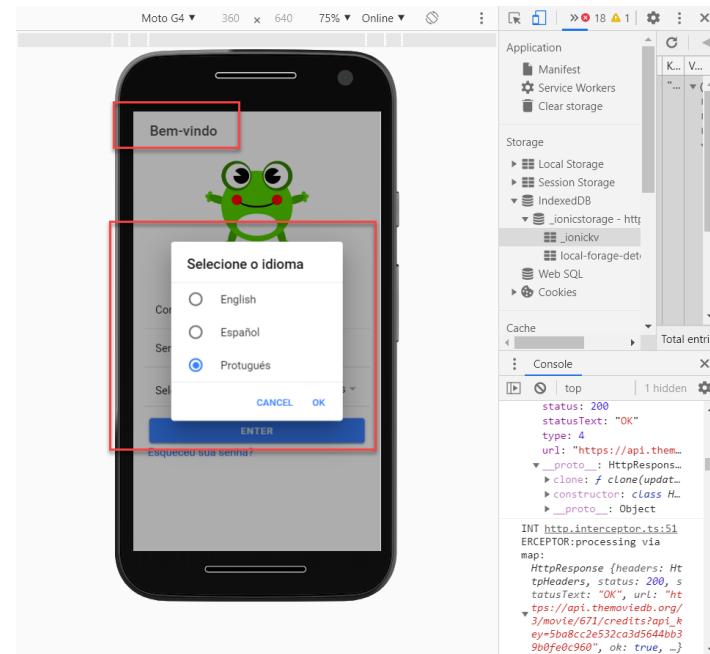
For this context, the option to purchase a movie is simulated, this causes the map window to be displayed to detect your location.



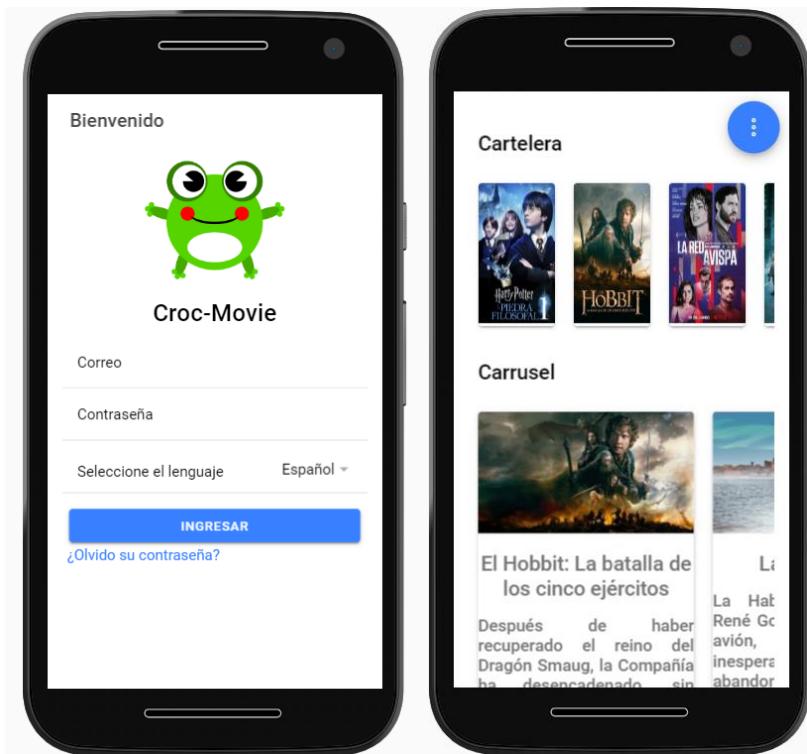
## 6- Multi-language support: English/Spanish.

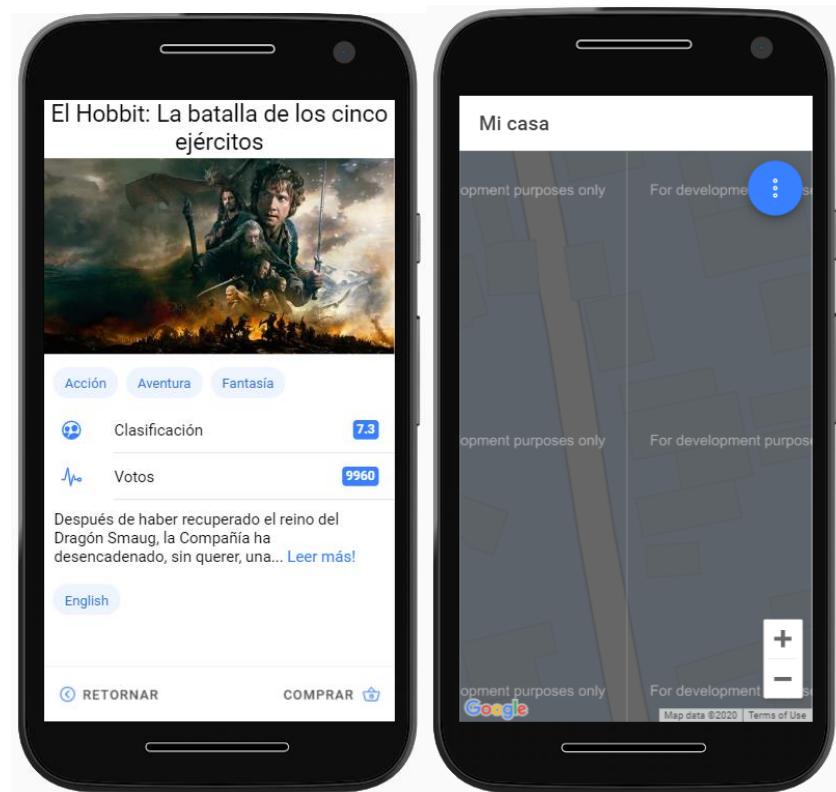
Install

- npm install --save @ngx-translate/core



The APP have a multiple Language Tranlate for all Pages.

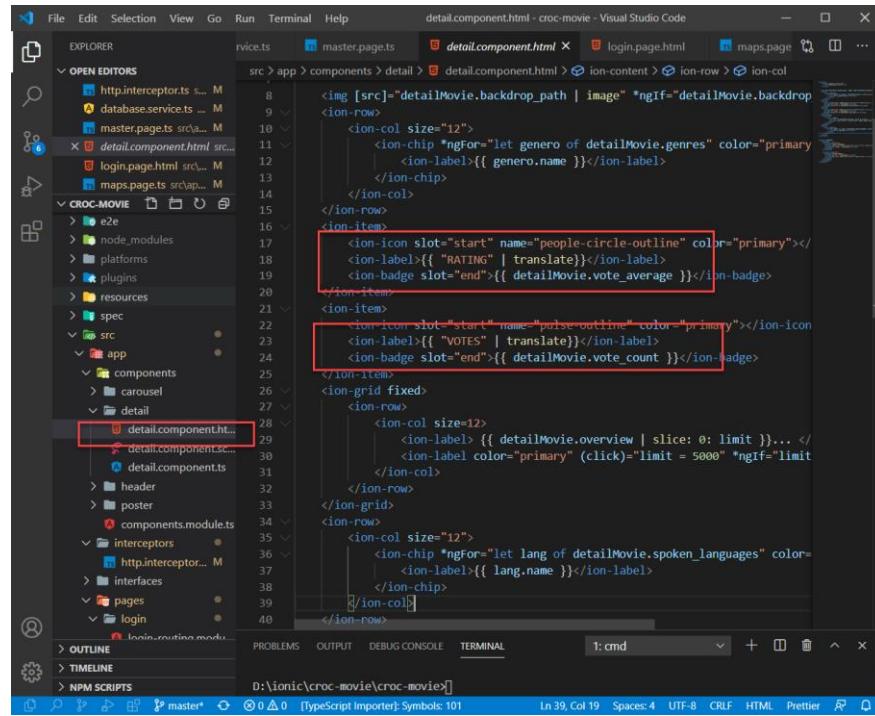




The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** On the left, it shows the project structure:
  - OPEN EDITORS**: login.page.html, maps.page.ts, en.json
  - CROC-MOVIE**: database.service.ts, database.service.spec.ts, movies.service.ts, movies.service.spec.ts, network.service.ts, network.service.spec.ts, translateapp.service.ts, translateapp.service.spec.ts, app-routing.module.ts, app.component.html, app.component.scss, app.component.ts, app.component.spec.ts, app.module.ts
  - assets**: i18n
    - en.json**
    - es.json
    - pt.json  - icon, croc.jpg, shapes.svg
- Code Editor:** The en.json file is open, showing the following JSON content:

```
1  [{"2   "WELCOME": "Welcome",3   "EMAIL": "Email",4   "PASSWORD": "Password",5   "SIGNIN": "Sing In",6   "FORGOT": "Forgot Password?",7   "LANGUAGE": "Select language",8   "POSTER": "Poster",9   "CAROUSEL": "Carousel",10  "RATING": "Rating",11  "VOTES": "Votes",12  "RETURN": "Return",13  "BUY": "Buy",14  "MAPS": "My House"}]
```
- Bottom Status Bar:** D:\ionic\croc-movie\croc-movie> [TypeScript Importer] Symbols: 101 CRLF JSON Prettier

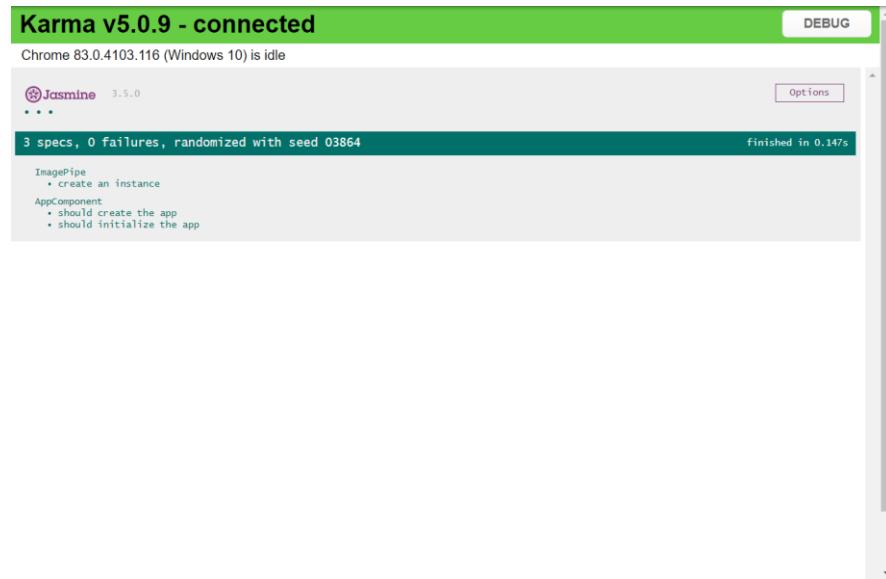


```
src > app > components > detail > detail.component.html > ion-content > ion-row > ion-col
  8   <img [src]="{{detailMovie.backdrop_path | image" *ngIf="detailMovie.backdrop
  9   <ion-row>
 10  <ion-col size="12">
 11    <ion-chip *ngFor="let genero of detailMovie.genres" color="primary
 12      <ion-label>{{ genero.name }}</ion-label>
 13    </ion-chip>
 14  </ion-col>
 15  </ion-row>
 16  <ion-item>
 17    <ion-icon slot="start" name="people-circle-outline" color="primary"></ion-icon>
 18    <ion-label>{{ "RATING" | translate}}</ion-label>
 19    <ion-badge slot="end">{{ detailMovie.vote_average }}</ion-badge>
 20  </ion-item>
 21  <ion-item>
 22    <ion-icon slot="start" name="pulse-outline" color="primary"></ion-icon>
 23    <ion-label>{{ "VOTES" | translate}}</ion-label>
 24    <ion-badge slot="end">{{ detailMovie.vote_count }}</ion-badge>
 25  </ion-item>
 26  <ion-grid fixed>
 27    <ion-row>
 28      <ion-col size="12">
 29        <ion-label>{{ detailMovie.overview | slice: 0: limit }}... </ion-label>
 30        <ion-label color="primary" (click)="limit = 5000" *ngIf="limit
 31          </ion-col>
 32    </ion-row>
 33    <ion-row>
 34      <ion-col size="12">
 35        <ion-chip *ngFor="let lang of detailMovie.spoken_languages" color=
 36          <ion-label>{{ lang.name }}</ion-label>
 37        </ion-chip>
 38    </ion-col>
 39  </ion-row>
```

For this solution I have used to Library Translator of Angular and implement via pipe this service

## -7 Add unit tests

For this task I have used Jasmine and Karma, configuring a module on APP



```
import { ImagePipe } from './image.pipe';

describe('ImagePipe', () => {
  it('create an instance', () => {
    const pipe = new ImagePipe();
    expect(pipe).toBeTruthy();
    expect(pipe).toEqual(!null);
  });
});
```

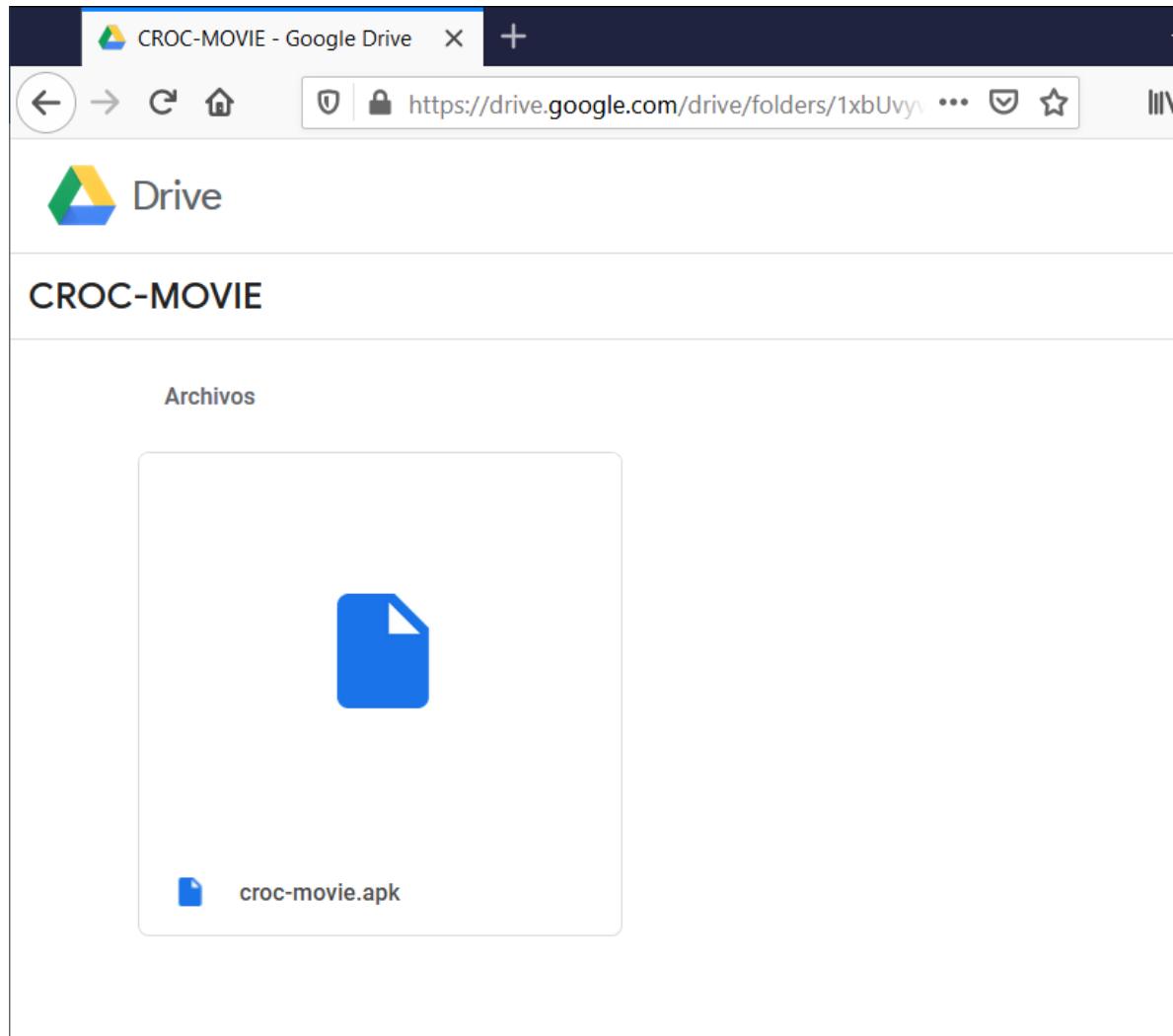
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure with files like `http.interceptor.ts`, `interfaces.ts`, `login.page.ts`, `maps.page.ts`, `master.page.ts`, `environment.prod.ts`, `image.pipe.spec.ts` (highlighted with a red box), and `environment.ts`.
- Editor (Center):** Displays the content of `image.pipe.spec.ts` with the test code highlighted by a red box.
- Terminal (Bottom):** Shows the command `ionic cordova build android` being run and completed successfully.

## DELIVERABLE

- APK uploaded and shared in Google Drive folder.

<https://drive.google.com/drive/folders/1xbUvyv3uggsO0zTA5AVr7vy42quCyZF2?usp=sharing>



- iOS: If you have MAC, screenshots of the app in the emulator. (optional)

## COMMANDS

- ionic cordova platform add android
- ionic cordova build android

- Source code versioned in a public Github repo. (Use the gitflow model: optional).

The screenshot shows a GitHub repository page for 'alexanderbismark007 / croc-movie'. The repository has 5 commits, 1 branch, and 0 tags. Recent activity includes commits from 'alexanderbismark007' and 'f976ee9' 1 hour ago. The repository structure includes 'documentation', 'e2e', 'resources', 'spec/support', 'src', '.gitignore', 'LICENSE', 'README.md', 'angular.json', and 'browserslist'. The 'About' section provides information about the project, including its purpose (Repo for project with https://www.themoviedb.org using IONIC 4 and Angular 9), a 'Readme' link, and an 'MIT License' link. Other sections include 'Releases' (No releases published, Create a new release), 'Packages' (No packages published, Publish your first package), and a 'Settings' link.

### - Document the environment config and libraries in the Readme file.

The screenshot shows the 'README.md' file content. It starts with a heading 'CROC-movie'. Below it, there is a detailed list of tools and configurations used in the project:

- Repo for project with <https://www.themoviedb.org> using IONIC and Angular
- TOOLS FOR PROJECT Google Chrome <https://www.google.com/chrome/>
- Nodejs <https://nodejs.org/es/> CMD: npm -v
- TypeScript <https://www.typescriptlang.org/> CMD: npm install -g typescript
- AngularCLI <https://cli.angular.io/> CMD: npm install -g @angular/cli • ng new my-dream-app • cd my-dream-app • ng serve
- Postman <https://www.getpostman.com/>
- Visual Studio Code <https://code.visualstudio.com/> • AB Html Formatter • Activitus Bar • Angular Snippets • Angular Language Service • ionic Snippets • Paste JSON as Code • TypeScript Importer
- Ionic y Cordova <https://ionicframework.com/> CMD: npm install -g ionic CMD: npm install -g cordova
- Android Studio <https://developer.android.com/studio/>

### To consider (optional):

- ### - Comments in the code

**COMPODOC** (Tool for document code)

CMD: npm install --save-dev @compodoc/compo

Add in package.json section scripts

```
"compodoc": "./node_modules/.bin/compodoc -p tsconfig.json -w -s"
```

CMD: npm run compodoc

```
ls npm
D:\ionic\croc-movie>npm run compodoc

> croc-movie@0.0.1 compodoc D:\ionic\croc-movie
> compodoc -p tsconfig.json -w -s

compodac

1.1.11

TypeScript version used by Compodoc : 2.9.1
TypeScript version of current project : 3.8.3
Node.js version : v12.18.0
Operating system : Windows 10

[11:11:58] No configuration file found, switching to CLI flags.
[11:11:58] Using tsconfig file : D:\ionic\croc-movie\tsconfig.json
[11:11:58] Including : D:\ionic\croc-movie\.gitignore
[11:11:58] Including : D:\ionic\croc-movie\angular.json
[11:11:58] Including : D:\ionic\croc-movie\browserslist
[11:11:58] Including : D:\ionic\croc-movie\ionic.config.json
[11:11:58] Including : D:\ionic\croc-movie\karma.conf.js
[11:11:58] Including : D:\ionic\croc-movie\package-lock.json
[11:11:58] Including : D:\ionic\croc-movie\package.json
[11:11:58] Including : D:\ionic\croc-movie\tsconfig.app.json
[11:11:58] Including : D:\ionic\croc-movie\tsconfig.json
[11:11:58] Including : D:\ionic\croc-movie\tsconfig.spec.json
[11:11:58] Including : D:\ionic\croc-movie\tslint.json
[11:11:58] Including : D:\ionic\croc-movie\webpack.proractor.conf.js
[11:11:58] Including : D:\ionic\croc-movie\webpack.tsconfig.json
[11:11:58] Including : D:\ionic\croc-movie\src\global.scss
[11:11:58] Including : D:\ionic\croc-movie\src\index.html
```

Access to: <http://127.0.0.1:8080/index.html>

croc-movie documentation

Type to search

Getting started

- Overview
- README
- LICENSE

Dependencies

Modules

Classes

Injectables

Interceptors

Interfaces

Miscellaneous

Routes

Documentation coverage

Documentation generated using

## Overview

The architecture diagram illustrates the system's flow. It starts with a central **Movie** component, which interacts with **Rating** and **Actor** components via **Provides** and **Exports** connections. These components then feed into a series of **RatingPipe**, **ActorPipe**, and **RatingActorPipe** pipes. These pipes connect to **RatingMapper**, **ActorMapper**, and **RatingActorMapper** modules, which finally output to **RatingResponse** and **ActorResponse** components.

Legend:

- Declarations (Yellow)
- Module (Green)
- Mapper (Blue)
- Provides (Orange)
- Exports (Red)

Zoom in | Reset | Zoom out

10 Modules

8 Components

4 Injectables

1 Pipe

10 Interfaces

10 Routes

**IMPORTANT:**

As the project progresses you can generate the documentation at any time, the important thing is to follow the COMPODOC rules (<https://compodoc.app/guides/demo.html>)

## - Proper error handling with custom exceptions.

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help
- Editor Title:** http.interceptor.ts - croc-movie - Visual Studio Code
- Explorer:** Shows the project structure under "CROC-MOVIE". The file "http.interceptor.ts" is selected and highlighted in blue. A red box highlights the "interceptors" folder and its contents.
- Code Editor:** Displays the content of "http.interceptor.ts". The code implements an HTTP interceptor to add an API key to requests and handle errors. A red box highlights the "catchError" block where an error response is logged and then thrown.
- Terminal:** Shows the command "D:\ionic\croc-movie\croc-movie>[]"
- Bottom Status Bar:** master\*, 0 △ 0 [TypeScript Importer]: Symbols: 101, Ln 14, Col 68, Spaces: 2, UTF-8, CRLF, TypeScript 3.9.4, Prettier

```
src > app > interceptors > http.interceptor.ts > ...
  33  switch(token) {
  34    let newHeaders = request.headers;
  35    newHeaders = newHeaders.append('API-KEY', environment.apiKey);
  36    request = request.clone(
  37      {
  38        //url: environment.urlDetails+/discover/movie?api_key=${environment.apiKey}
  39        setHeaders: { Authorization: `API-KEY ${environment.apiKey}` }
  40      });
  41      console.log('INTERCEPTOR HEADER',request);
  42    return next
  43      .handle(request)
  44      .pipe(
  45        tap((event: HttpEvent<any>) => {
  46          if (event instanceof HttpResponse) {
  47            console.log('INTERCEPTOR:processing via tap:', event);
  48          }
  49        }),
  50        map((event: HttpEvent<any>) => {
  51          if (event instanceof HttpResponse) {
  52            console.log('INTERCEPTOR:processing via map:', event);
  53          }
  54          return event;
  55        }),
  56        catchError((error: HttpErrorResponse) => {
  57          console.error("INTERCEPTOR ERR.HttpErrorResponse:",error);
  58          return throwError(error);
  59        })
  60      );
  61    }
  62  );
  63
  64  async presentErrorToast(msg) {
  65    const toast = await this.toastCtrl.create({
```

- ReactiveX

File Edit Selection View Go Run Terminal Help http.interceptor.ts - croc-movie - Visual Studio Code

EXPLORER OPEN EDITORS http.interceptor.ts database.service.ts master.page.ts network.service.ts

CROC-MOVIE

detail.component.html detail.component.scss detail.component.ts header poster components.module.ts interceptors http.interceptor.ts interfaces pages login login-routing.module.ts login.module.ts login.page.html login.page.scss login.page.ts login.page.spec.ts maps maps-routing.module.ts maps.module.ts maps.page.html maps.page.scss

src > app > interceptors > http.interceptor.ts > ...

```
import { environment } from '../../environments/environment.prod';
import { Observable, throwError, from } from 'rxjs';
import { map, tap, catchError, switchMap } from 'rxjs/operators';

const TOKEN_KEY = 'BismarkToken';
@Injectable()

export class HttpConfigInterceptor implements HttpInterceptor {

  isLoading: boolean = false;
  constructor(public storage: storage,
    public loadingCtrl: LoadingController,
    private toastCtrl: ToastController) { }

  intercept(request: HttpRequest<any>,
    next: HttpHandler):
    Observable<HttpEvent<any>> {
      return from(this.storage.get(TOKEN_KEY))
        .pipe(
          switchMap(token => {
            let newHeaders = request.headers;
            newHeaders = newHeaders.append('API-KEY', environment.apiKey);
            request = request.clone(
              {
                //url: environment.urlDetails`/discover/movie?api_key=${environment.apiKey}
                setHeaders: { Authorization: `API-KEY ${environment.apiKey}` }
              );
            console.log('INTERCEPTOR HEADER',request);
            return next
              .handle(request)
              .pipe(
                tap((event: HttpEvent<any>) => {
                  if (event instanceof HttpResponse) {
                    if (event.status === 401) {
                      this.isLoading = true;
                      this.toastCtrl.show('Session Expired', 3000, 'bottom');
                      this.storage.remove(TOKEN_KEY);
                      this.isLoading = false;
                      this.router.navigate(['/login']);
                    }
                  }
                })
              )
            );
          })
        );
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: cmd + - ×

D:\ionic\croc-movie\croc-movie\[]

master\* ① 0 △ 0 [TypeScript Importer]: Symbols: 101 Ln 14, Col 68 Spaces: 2 UTF-8 CRLF TypeScript 3.9.4 Prettier