



<MOVIEMAN>

<ON OUR WEBSITE YOU CAN WATCH, RATE, COMMENT ON
VARIOUS FILMS>

FRONT END

- 1 `import { Injectable } from '@angular/core';`
`import { HttpClient, HttpClientModule } from '@angular/common/http';`
`import {Movie} from './movies';`
`import {Observable} from 'rxjs';`
`import {Commentary} from './commentary';`
- 2 Services : `genre.service`, `login.service`, `movie.service`.
- 3 Minimum 4 (`onclick`) events, which hits API: We have 4 onclick events that can change the API. Such as Event for "Like", "Dislike", "Share" and "Publish" buttons.
- 4 We used `[(ngModel)]` directives with and without additional settings.
- 5 Css files : We have css files in each component
- 6 Configured Routing module: In the `app-routing.module.ts` file we defined routes using the path properties
- 7 `ngFor`, `ngIf`: We used the `*ngFor` directive to display a list of movies. and the `*ngIf` directive for conditional display of elements
- 8 `@Injectable()` `export class JwtInterceptor implements HttpInterceptor { intercept(request: HttpRequest<any>, next: HttpHandler): Observable`

BACK END

- 1 Minimum 4 models : Genre,movie,user,comment
- 2 all 4 classes are defined as model Manager
- 3 Using ForeignKey, we built a relationship between genre and movie, commentary and movie
- 4 Serializer:UserSerializer,GenreSerializer are serializers based on the serializers.Serializer class MovieSerializer, CommentSerializer are serializers based on the serializers.ModelSerializer class
- 5 FBV drf view :genres_list,genres_movies and CBV APIview:movies_list, movies_detail, UsersListAPIView, UserDetailsAPIView, CommentsListAPIView, CommentDetailAPIView
- 6 path('login/token/', TokenObtainPairView.as_view(), name='token_obtain_pair'),
- 7 At least for 1 model CRUD operations : We created a crud operation for the movie model

READY WEBSITE

