Cinephill

Cinephill is a backend application built using Spring Boot 6 and Spring Security 3 that provides REST services for movie/ tv show search, user authentication, and notifications. The application interacts with external services and APIs, including AWS SSM, AWS KMS, SES, and The Movie Database API (TMDB).

Purpose

Cinephill is a movie search application that allows users to search for movies and TV shows by category, time frame, and rating. Users can also search for a movie's credits and cast information by entering the movie ID. Additionally, users can search for a movie based on a query parameter and customize their category settings to receive notifications for new TV show releases based upon their criterias.

Some Features

The following are the some features of Cinephill:

- Search for movies and TV shows by category, time frame, and rating
- Search for a movie's credits and cast information by entering the movie ID
- Search for a movie based on a query parameter e.g. movieName, actorName, special keyword etc.
- Customize category settings to receive notifications for new TV show releases
- User authentication and authorization using JWT tokens
- Private endpoints for authenticated users

Endpoints

Cinephill provides the following REST endpoints:

Public Endpoints

- GET ${..}/discover/tv - Returns a list of TV shows
- GET ${..}/movie/popular - Returns a list of popular movies
- GET ${..}/movie/{movieId} - Returns the details of a specific movie
- GET ${..}/movie/{movieId}/credits - Returns the credits for a specific movie
- GET ${..}/search/movie?query={query} - Returns a list of movies based on a search query
- GET ${..}/public/movies/character - Returns the character name by entering a movie name and an actor/actress's name
- GET ${..}/public/movies/popular - Returns a list of popular movies
- GET ${..}/public/movies/search - Returns a list of movies based on a search query
- GET ${..}/public/movies/{movieId} - Returns the details of a specific movie
- GET ${..}/public/movies/{movieId}/cast - Returns the cast for a specific movie

Private Endpoints

- GET ${..}/private - Returns private data for authenticated users
- GET ${..}/private/tv/search - Returns a list of TV shows based on a search query
- POST ${..}/private/users/{userId}/category - Adds a category setting to a user's profile

Authentication Endpoints

- POST ${..}/api/auth/login - Authenticates a user and returns a JWT token
- POST ${..}/api/auth/register - Registers a new user and returns a JWT token

Deployment

Cinephill can be deployed using Docker. Docker images can be built using The Paketo Buildpack for Maven automatically. Once the image is built, the application can be run using Docker Compose.

Security

Cinephill uses JWT tokens for user authentication and authorization. Users must authenticate to access private endpoints. Sensitive data, including db credentials and API keys, are stored securely using AWS SSM and AWS KMS.

Please note that parameters also kept in application.yml file to easy development.

Debugging

- Feign client for TMDb API: TMDb API requires the `api_key` query parameter for every request. Cinephill includes an interceptor that adds the `api_key` to every request made by the Feign client. This allows developers to test the application without embedding the API URL to the query parameter for each request.
- Logging: Cinephill includes extensive logging capabilities, including logging of request and response data for debugging purposes.
- Global Exception Handler: Cinephill includes a global exception handler that catches and handles exceptions thrown by the application. This helps developers to quickly identify and fix issues.
- Custom Error Details: Cinephill includes a custom error details class that provides timestamp, message, and path details for errors. This helps developers to quickly identify and diagnose issues.
- Aspects: Cinephill includes an aspect class that logs the method execution time and arguments. This helps developers to identify performance bottlenecks and optimize the application.

- ModelMapper: Cinephill uses ModelMapper to map between DTO and entity classes. This reduces the amount of boilerplate code required and improves the maintainability of the application.

Deployment

Cinephill includes the following deployment features for developers:

- Docker Images: Cinephill can be built as a Docker image using the `mvn spring-boot:build-image` command. Each image can be built with a special name or project ID for easy identification.
- AWS SSM: Cinephill uses AWS SSM to store sensitive data such as API keys and database credentials. This allows developers to securely store and retrieve this data during development and deployment.
- AWS KMS: Cinephill uses AWS KMS to encrypt sensitive data stored in AWS SSM. This provides an additional layer of security during deployment.
- Scheduled Jobs: Cinephill includes scheduled jobs for sending bi-weekly notifications to users. Developers can configure the frequency of these jobs and customize the email template.