

Spring Boot Task

Your task is to create a backend service with a database and API that supports the features needed for a mobile app or web app. Additionally, you need to implement an API key system to allow users to access the API without requiring full authentication.

****Important****

- You have 3 days to complete this part and upload your code to a repo of your choice. (GitHub or GitLab)
- The repository must be private and shared with me.
- Don't push the API key.
- Once you've completed the assignment, email your contact person with the repository link.

****Database Design****

Design and implement a database to store movie data, supporting the following features:

- A list of popular movies.
- The ability to search for movies by title.
- Detailed information about specific movies.

Your schema should include:

- Movies Table: ID, Title, Release Date, Poster URL, Overview, Genres, Rating, Runtime, Language.

Populate the database with sample movie records to simulate a real-world scenario. Add enough sample data to test pagination.

Create the following ****API endpoints****:

****Popular Movies****

- URL: /movies/popular
- Method: GET
- Purpose: Retrieve the top 50 popular movies.
- Query Parameters: page (optional) for pagination.
- Response:
 - Movie ID
 - Title
 - Release Date
 - Poster URL
 - Average Rating

****Search for Movies****

- URL: /movies/search
- Method: GET
- Query Parameters: query (required), sort_by (optional), and filter (optional).
- Purpose: Search for movies by title, and optionally sort or filter results.
- Response:
 - Movie ID
 - Title
 - Release Date
 - Poster URL
 - Average Rating

****Movie Details****

- URL: /movies/{id}
 - Method: GET
 - Purpose: Retrieve detailed information about a specific movie.
 - Response:
 - Title
 - Release Date
 - Full Poster URL
 - Overview
 - Genres
 - Average Rating
 - Runtime
 - Language
-

****API Key System****

- Implement an API key system to control access to your endpoints.
- Users should include their API key as a query parameter in each request (e.g., /movies/popular?api_key=YOUR_API_KEY).
- Validate the API key for each request and return an appropriate error if it's missing or invalid.
- Do not implement an API key generation system. Simple hard-coded keys are enough.

****Technical Requirements****

- Use a backend framework Spring Boot.
- Write clean, maintainable, and scalable code.
- Implement error handling for invalid inputs and failed requests.
- Write tests for critical functionality, including:
 - Unit tests for endpoints and database queries.
 - Integration tests to validate end-to-end API functionality.

****Submission Guidelines****

- Push your code to a public repository.
- Include a README.md file with:
 - Setup instructions.
 - Details of the API key system.
 - A description of your database schema.
 - How to test the API.
- Email your contact person with the repository link when you're done.

****Optional (Bonus Points)****

- Deploy the API to a cloud service (e.g., AWS, Heroku, etc) and include the link in the README.