**Video Streaming Documentation**

**1. How to Run the Program Using Maven**

**Prerequisites:**

Ensure that you have Java 17 or higher installed. You can verify this by running:

Make sure Maven is installed on your system.

**Steps to Run the Program:**

**Step 1: Clone the Repository**

Clone the repository from your source control (e.g., GitHub or GitLab):

git clone https://github.com/chetanpatidar1006/VideoStreamingRepo.git

**Step 2: Navigate to the Project Folder**

**Step 3: Build the Project**

To build the project and download all necessary dependencies, run the following Maven command:

mvn clean install

This command will: Generate a .jar file inside the target/ directory.

**Step 4: Run the Application**

Once the project is built, run it with:

mvn spring-boot:run

**Step 5: Verify the Application**

After the application starts, open your browser or use Postman/cURL to check if it's running at:

http://localhost:8080

**cURL Commands for Testing API Endpoints**

1. **Publish a Video** (POST /video/v1)

curl -X POST http://localhost:8080/video/v1 \

-H "Content-Type: application/json" \

-d '{

"title": "Inception",

"director": "Christopher Nolan",

"cast": "Leonardo DiCaprio",

"genre": "Sci-Fi",

"runningTime": 148,

"yearOfRelease": "2010",

"synopsis": "A thief who steals corporate secrets...",

"deListed": false

}'

**2. Update Video Metadata** (PUT /video/v1/{id})

curl -X PUT http://localhost:8080/video/v1/1 \

-H "Content-Type: application/json" \

-d '{

"title": "Inception Updated",

"director": "Christopher Nolan",

"cast": "Leonardo DiCaprio",

"genre": "Sci-Fi",

"runningTime": 150,

"yearOfRelease": "2010",

"synopsis": "Updated synopsis"

}'

1. **Load Video Details** (GET /video/v1/{id})

curl -X GET <http://localhost:8080/video/v1/1>

1. **Play Video** (POST /video/v1/play/{id})  
     
    curl -X POST <http://localhost:8080/video/v1/play/1>

5. **List All Videos** (GET /video/v1)  
  
 curl -X GET http://localhost:8080 /video/v1

**6. Search Videos by Query Parameters** (GET /video/v1/search)

We can Search By these parameters   
 1. directorName

2. title

3. Cast

4. genre

curl -X GET <http://localhost:8080/video/v1/search?directorName=Christopher%20Nolan&title=Inception>

**7. Get Video Engagement Stats** (GET /video/v1/engagement/{id})  
  
 curl -X GET http://localhost:8080 /video/v1/engagement/1

**8. Delist Video** (DELETE /video/v1/{id})

curl -X DELETE <http://localhost:8080/api/videos/v1/1>

**Details About H2 Database**

H2 Database is an in-memory SQL database that stores data temporarily during the application's runtime. This is ideal for development and testing environments because the data is wiped out when the application is stopped.

To access H2 db while running the application user given URL in browser

localhost:8080/h2-console

username – sa

password - password

**Test Cases Overview for Video Management API**

The test cases focus on **positive scenarios** to ensure the correct functionality of the Video Management API. We have written tests to cover the main API functionalities, including publishing, updating, delisting, and retrieving videos. These tests verify that each endpoint behaves as expected when provided with valid inputs, ensuring smooth operations for video management and engagement tracking. Negative cases and error handling have not been covered in this suite, as the focus is on verifying expected behaviour.Top of Form