Title of Project

Film Finder

Application Description

This application will allow users to find movies given a criteria. By using a NoSQL deployment, we can scale horizontally to allow more data storage. Additionally, it is easier to query data because the database is schemaless.

Use Cases

1. Admin insert movie
2. Find movie given name
3. Find movie given date
4. Find movie given genre
5. Find movie given userID
6. Find movie given range of rating aggregate as average
7. Find ratings given movie and timestamp range
8. Admin deletes rating
9. User adds in their own rating on a movie
10. User update own rating on a movie
11. Find movie given tags
12. Find tags given movie
13. Admin deletes tags
14. User adds tags on a movie
15. User update tags on a movie
16. Find a tag given userID
17. Find movie given tag
18. Find movie given tag and relevance range
19. Find tag given genre

Description of dataset

Dataset contains movie information. It has a list of movies, associate user-created tags for each movie, a list of genomes tags describing the movie, and ratings given by users for each movie.

Link to dataset: https://grouplens.org/datasets/movielens/

Data Wrangling

The CSV file has genres separated by pipe symbols, so we will convert them into an array of Strings.

High Level Language and Driver

High Level Language: JavaScript

Driver: Mongoose

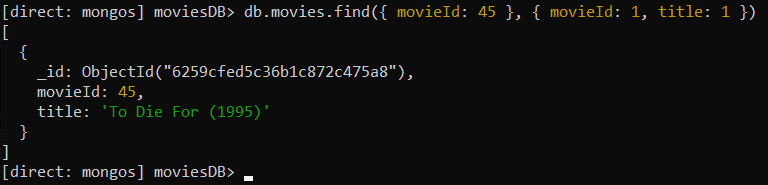
Schema Description

This dataset consists of four collections:

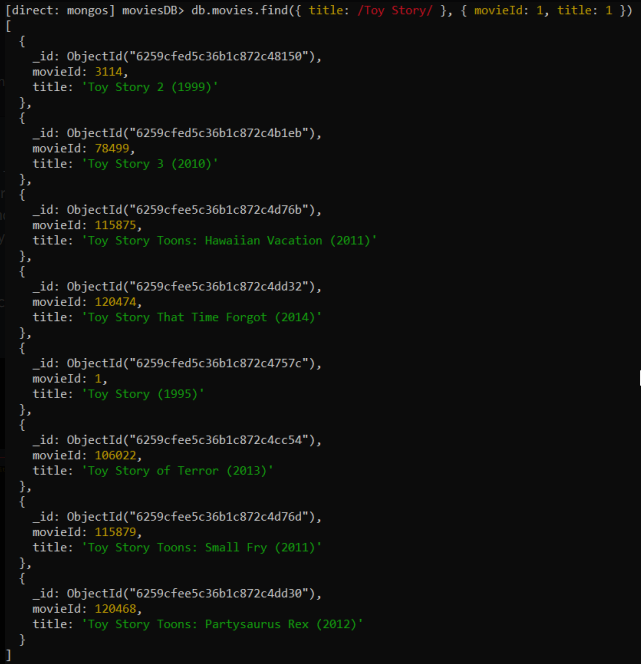
* Movie (lists of movies):
  + movieid: number
  + title: String
  + Genre: String[]
* Tags (user created tags describing a movie):
  + userID: number
  + movieID: number
  + tag: String
  + timestamp: date
* Genome (genes of movie of how related each tag is to movie):
  + movieId: number
  + tag: String
  + relevance: number
* Ratings (rating given by a user for a movie):
  + userID: number
  + movieID: number
  + rating: number
  + timestamp: date

Test Queries

Finding document by movieId:



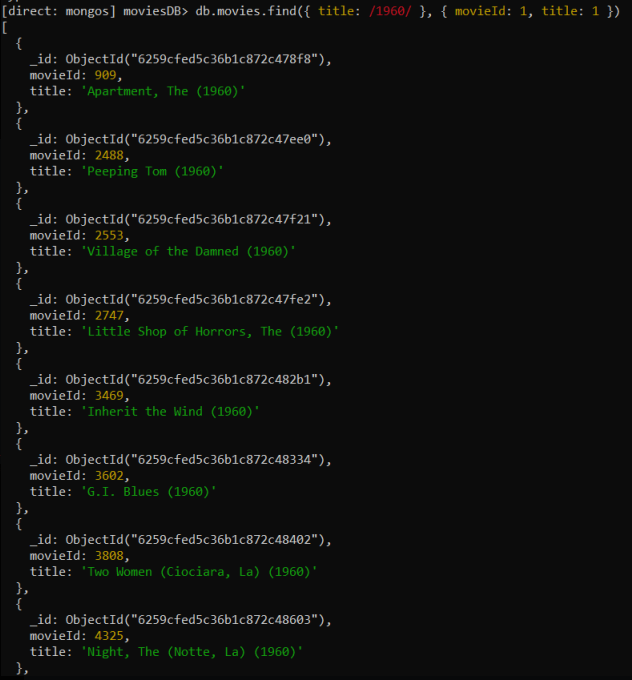
Finding document by title:



Finding document by genre:



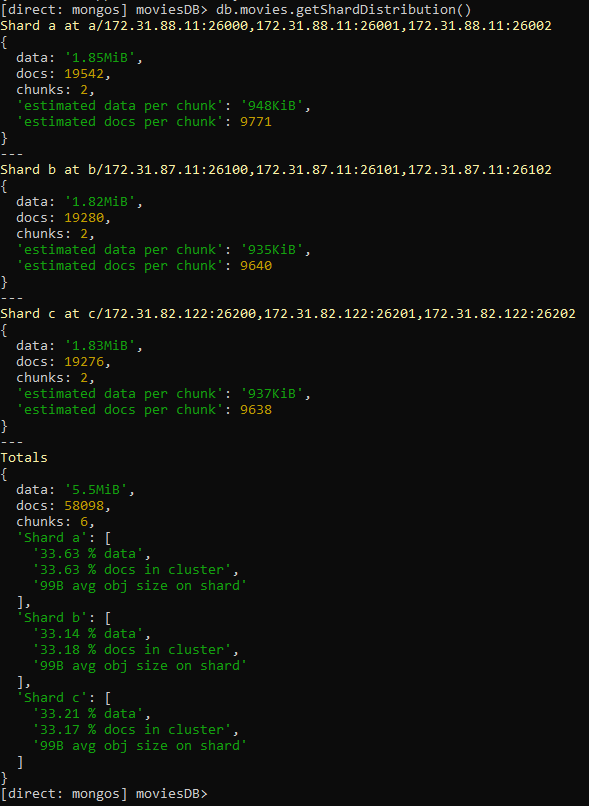
Find movie by year released:



Find document by array of movieId:



End-To-End Connection

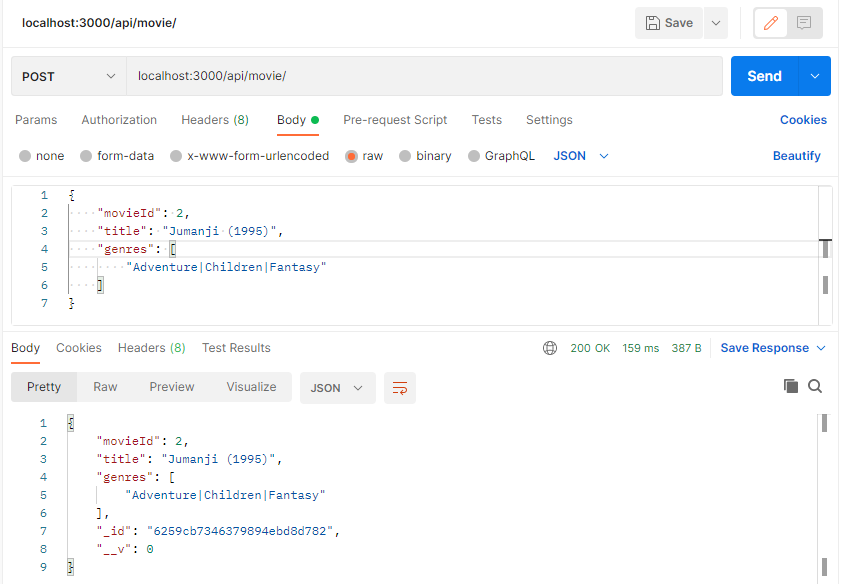


Read Data

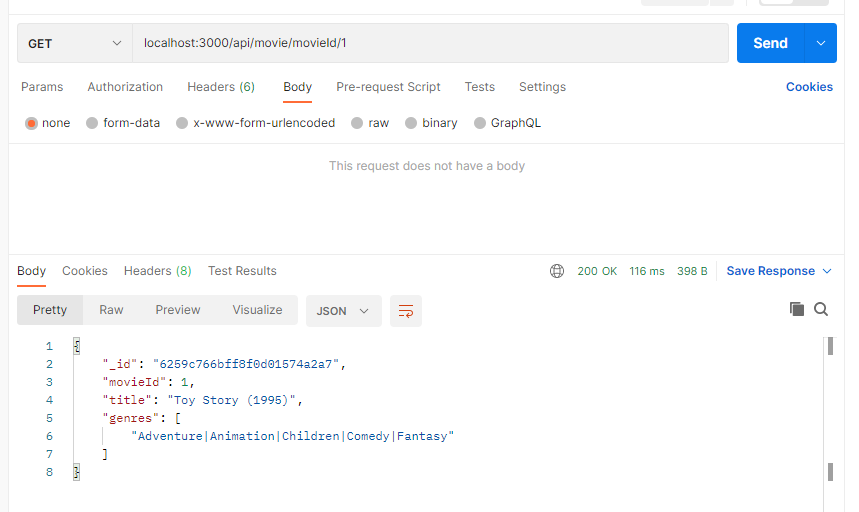


CRUD operations

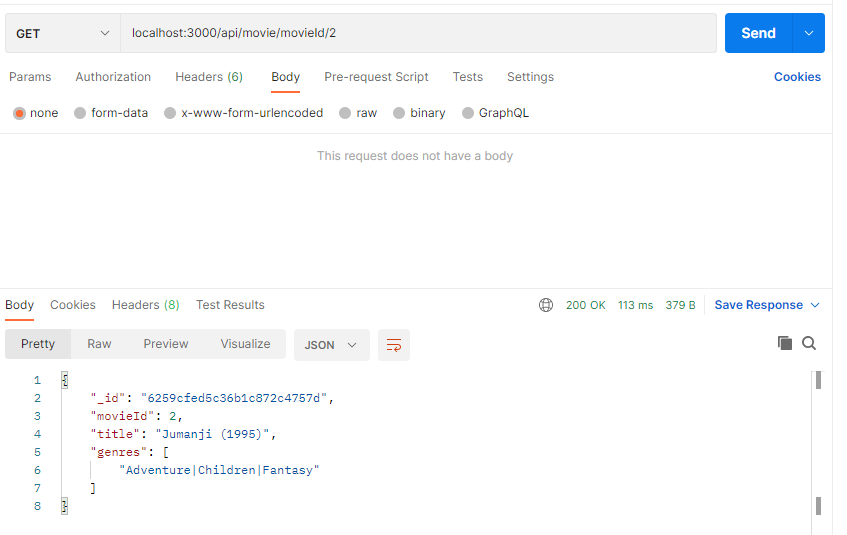
Create

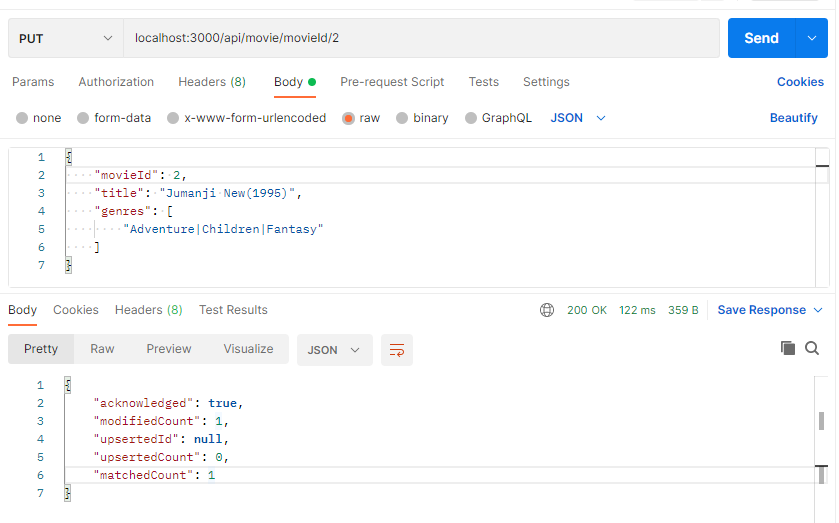


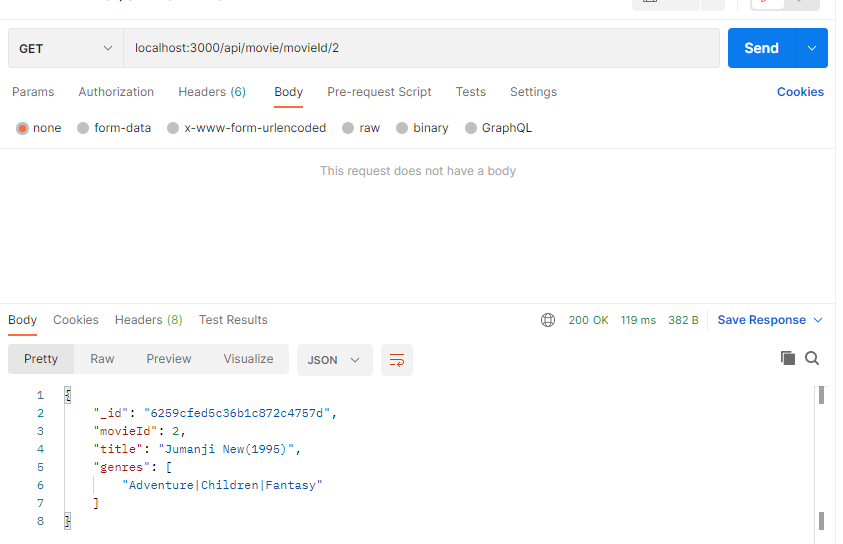
Read



Update







Delete

