

Integrating with CinemaBooking's Movie Database

I will now try to integrate with the CinemaBooking's Movie Database using mongosh and by following the integration guide provided by the exposee.

After installing mongosh im ready to connect to the database using:

```
user@Users-MacBook-Pro ~ % mongosh --host localhost -u integrator -p securepassword --authenticationDatabase CinemaBooking

Current Mongosh Log ID: 693e755f91f33cd49c973da3
Connecting to:      mongodb://<credentials>@localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=CinemaBooking&app
Name=mongosh+2.4.0
Using MongoDB:      7.0.16
Using Mongosh:      2.4.0
mongosh 2.5.10 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://www.mongodb.com/docs/mongosh-shell/

test> use CinemaBooking
switched to db CinemaBooking
```

Trying to read members does indeed confirm that i don't have the necessary permissions to interact with the member collection:

```
CinemaBooking> db.members.find({})
MongoServerError[Unauthorized]: not authorized on CinemaBooking to execute command
-4828-8a8e-970f17eced3a") }, $db: "CinemaBooking" }
CinemaBooking> █
```

Lets use the examples provided in the guide to query some data from movies. First out i want to find the year of release for Pulp Fiction and return it:

```
CinemaBooking> db.movies.find(
...   { Title: "Pulp Fiction" },
...   { "Year": 1, _id: 0 }
[... ]
[ { Year: 1994 } ]
CinemaBooking> █
```

Lets try and find Travolta's first name (Smallest unit):

```
CinemaBooking> db.movies.aggregate([
...   { $match: { Title: "Pulp Fiction" } },
...   { $unwind: "$Actors" },
...   { $match: { "Actors.LastName": "Travolta" } },
...   { $project: { _id: 0, FirstName: "$Actors.FirstName" } }
... ])
[...
[ { FirstName: 'John' } ]
CinemaBooking>
```

Lets try and insert a new movie:

```
CinemaBooking> db.movies.insertOne({
...   Title: "El Jefe Returns",
...   Summary: "In a distant future, El Jefe returns from the future to wreck havoc against those that crossed him in the future",
...   Runtime: 120,
...   IMDBRating: 9.1,
...   Year: 2008,
...   Director: {
...     FirstName: "John",
...     LastName: "Doe"
...   },
...   Genres: [],
...   Actors: []
... })
{
  acknowledged: true,
  insertedId: ObjectId('693f71a8e55befbed64d4689')
}
```

Let's update the director's last name to 'Bro':

```
CinemaBooking> db.movies.updateOne(
...   { Title: "El Jefe Returns" }, // filter by title
...   { $set: { "Director.LastName": "Bro" } } // set lastname to Doe
[... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
CinemaBooking> db.movies.find({Title: "El Jefe Returns"})
[
  {
    _id: ObjectId('693f71a8e55befbed64d4689'),
    Title: 'El Jefe Returns',
    Summary: 'In a distant future, El Jefe returns from the future to wreck havoc against those that crossed him in the future',
    Runtime: 120,
    IMDBRating: 9.1,
    Year: 2008,
    Director: { FirstName: 'John', LastName: 'Bro' },
    Genres: [],
    Actors: []
  }
]
```

After performing the update and query the movie, i can see that the last name was updated correctly.

Lets add an actor to the movie:

```
CinemaBooking> db.movies.updateOne(
...   { Title: "El Jefe Returns" },
...   {
...     $push: {
...       Actors: { FirstName: "Chuck", LastName: "Norris" }
...     }
...   }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
CinemaBooking> db.movies.find({Title: "El Jefe Returns"})
[
  {
    _id: ObjectId('693f71a8e55befbed64d4689'),
    Title: 'El Jefe Returns',
    Summary: 'In a distant future, El Jefe returns from the future to wreck havoc against those that crossed him in the future',
    Runtime: 120,
    IMDBRating: 9.1,
    Year: 2008,
    Director: { FirstName: 'John', LastName: 'Bro' },
    Genres: [],
    Actors: [ { FirstName: 'Chuck', LastName: 'Norris' } ]
  }
]
CinemaBooking> 
```

Let's change the actor's last name to 'Bro':

```
CinemaBooking> db.movies.updateOne(
...   { Title: "El Jefe Returns", "Actors.LastName": "Norris" },
...   { $set: { "Actors.$LastName": "Bro" } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
CinemaBooking> db.movies.find({Title: "El Jefe Returns"})
[
  {
    _id: ObjectId('693f71a8e55befbed64d4689'),
    Title: 'El Jefe Returns',
    Summary: 'In a distant future, El Jefe returns from the future to wreck havoc against those that crossed him in the future',
    Runtime: 120,
    IMDBRating: 9.1,
    Year: 2008,
    Director: { FirstName: 'John', LastName: 'Bro' },
    Genres: [],
    Actors: [ { FirstName: 'Chuck', LastName: 'Bro' } ]
  }
]
CinemaBooking> 
```