Exercises 2.10a

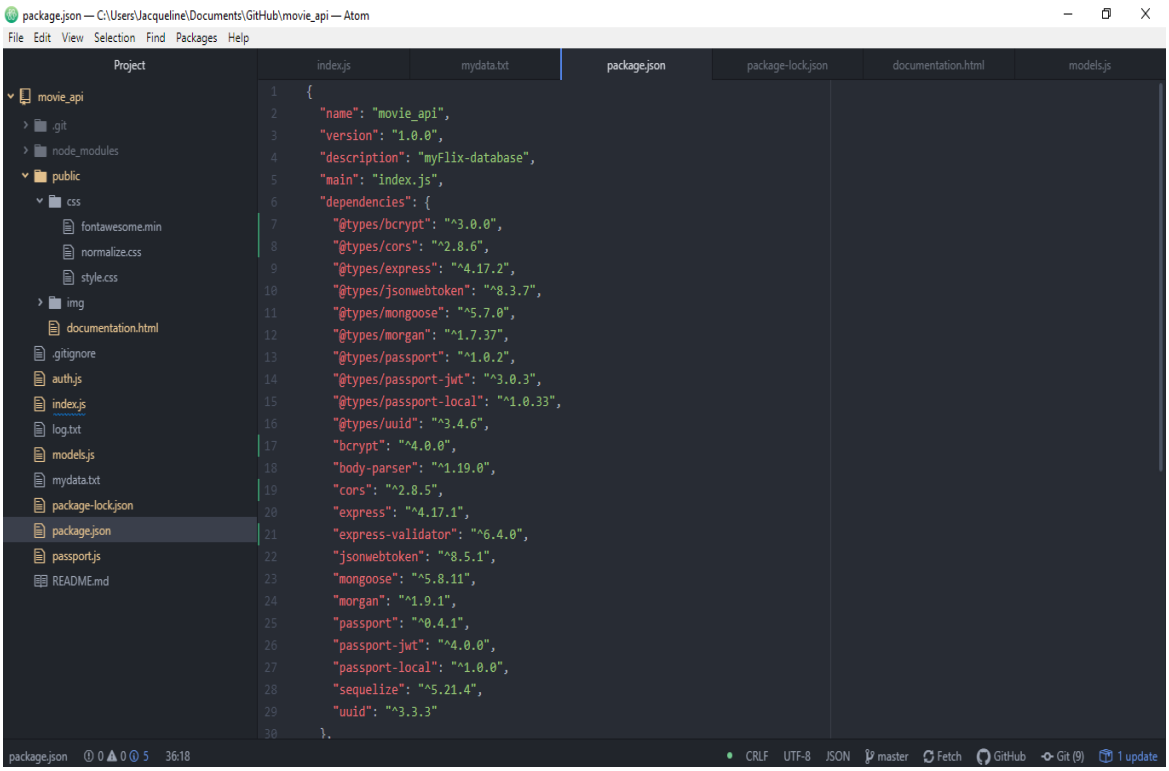
Installed CORS, bcrypt and express-validator via npm (git bash)

MINGW32:/c/Users/Jacqueline/Documents/GitHub/movie_api

```
Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ npm install cors
+ cors@2.8.5
added 2 packages from 2 contributors and audited 411 packages in 3.185s

1 package is looking for funding
  run `npm fund` for details

found 0 vulnerabilities


Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ npm install bcrypt

> bcrypt@4.0.0 install C:\Users\Jacqueline\Documents\GitHub\movie_api\node_modules\bcrypt
> node-pre-gyp install --fallback-to-build

node-pre-gyp WARN Using needle for node-pre-gyp https download
[bcrypt] Success: "C:\Users\Jacqueline\Documents\GitHub\movie_api\node_modules\bcrypt\lib\binding\napi-v3\bcrypt_lib.node" is installed via remote
+ bcrypt@4.0.0
added 63 packages from 90 contributors and audited 510 packages in 15.614s

2 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities


Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ npm install @types/cors
+ @types/cors@2.8.6
added 1 package from 1 contributor and audited 524 packages in 3.086s

2 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities


Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ npm install @types/bcrypt
+ @types/bcrypt@3.0.0
added 1 package from 3 contributors and audited 525 packages in 8s

2 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

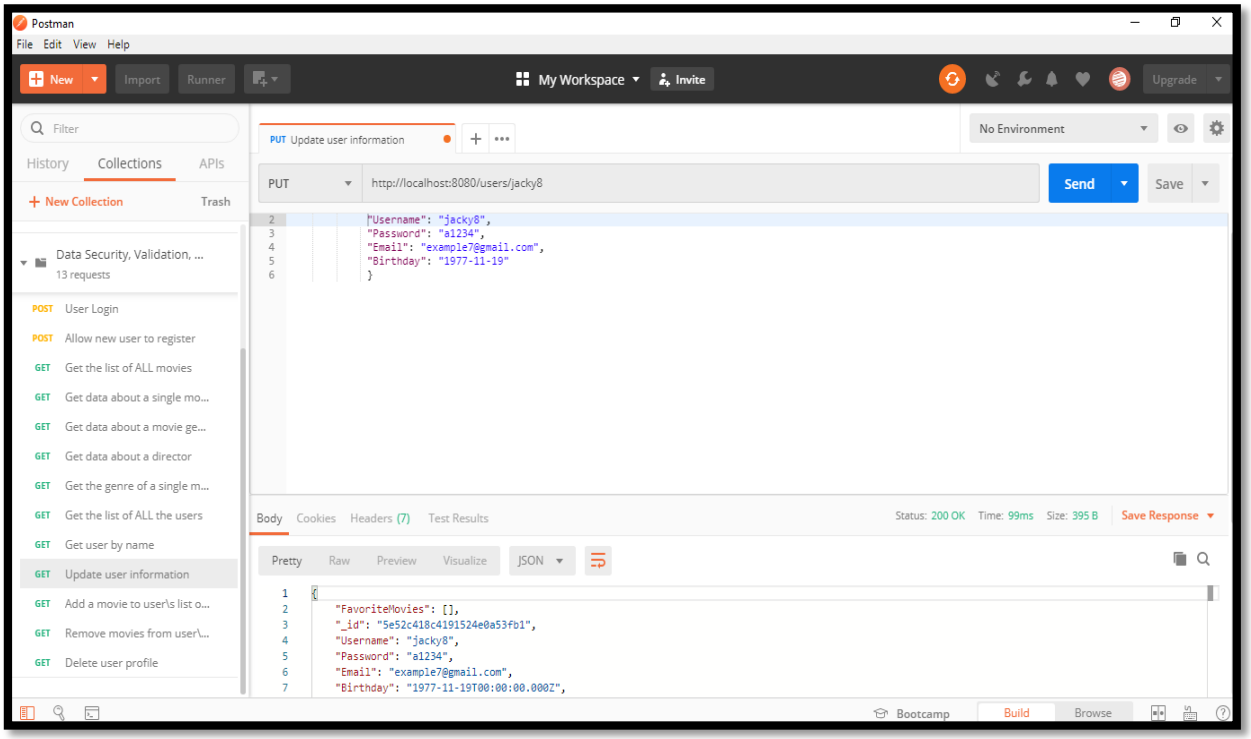
```
Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ npm install express-validator
+ express-validator@6.4.0
added 2 packages from 5 contributors and audited 528 packages in 4.684s

2 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Updated package.json:

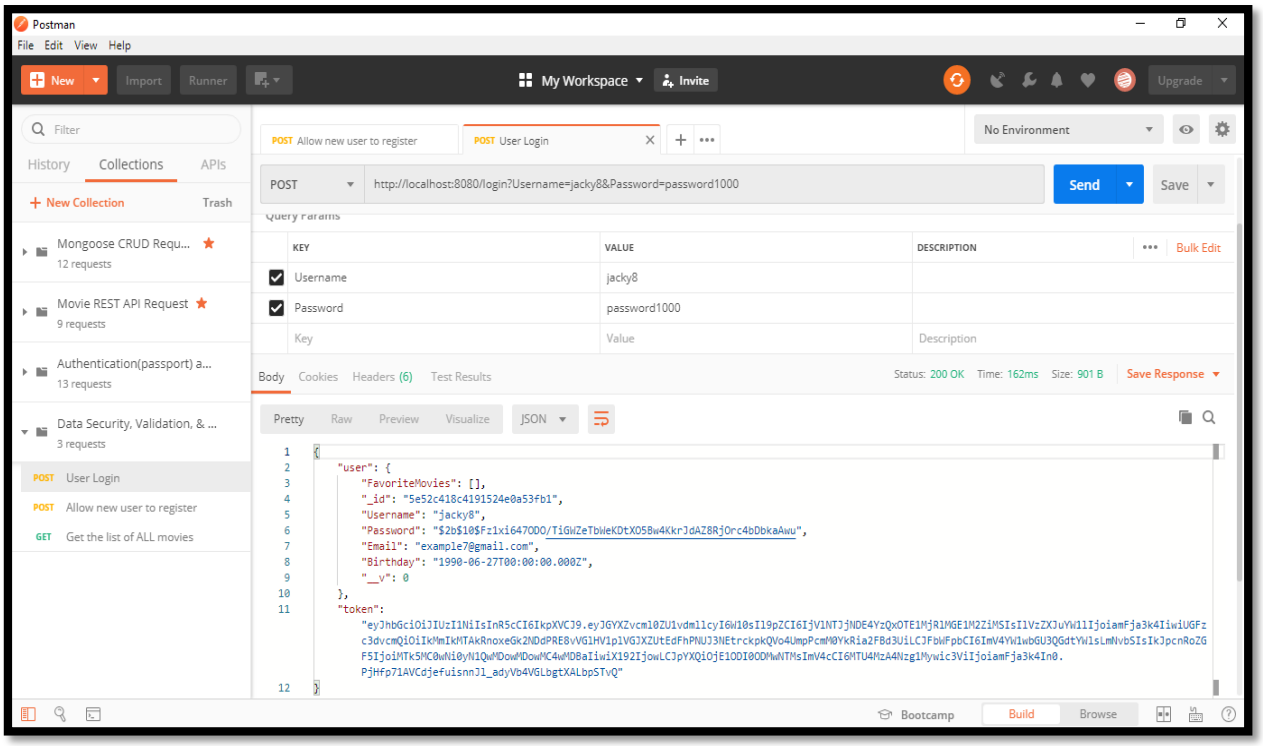


Postman screenshots (testing code before deploying via Heroku and MongoDB Atlas)

New User Registration

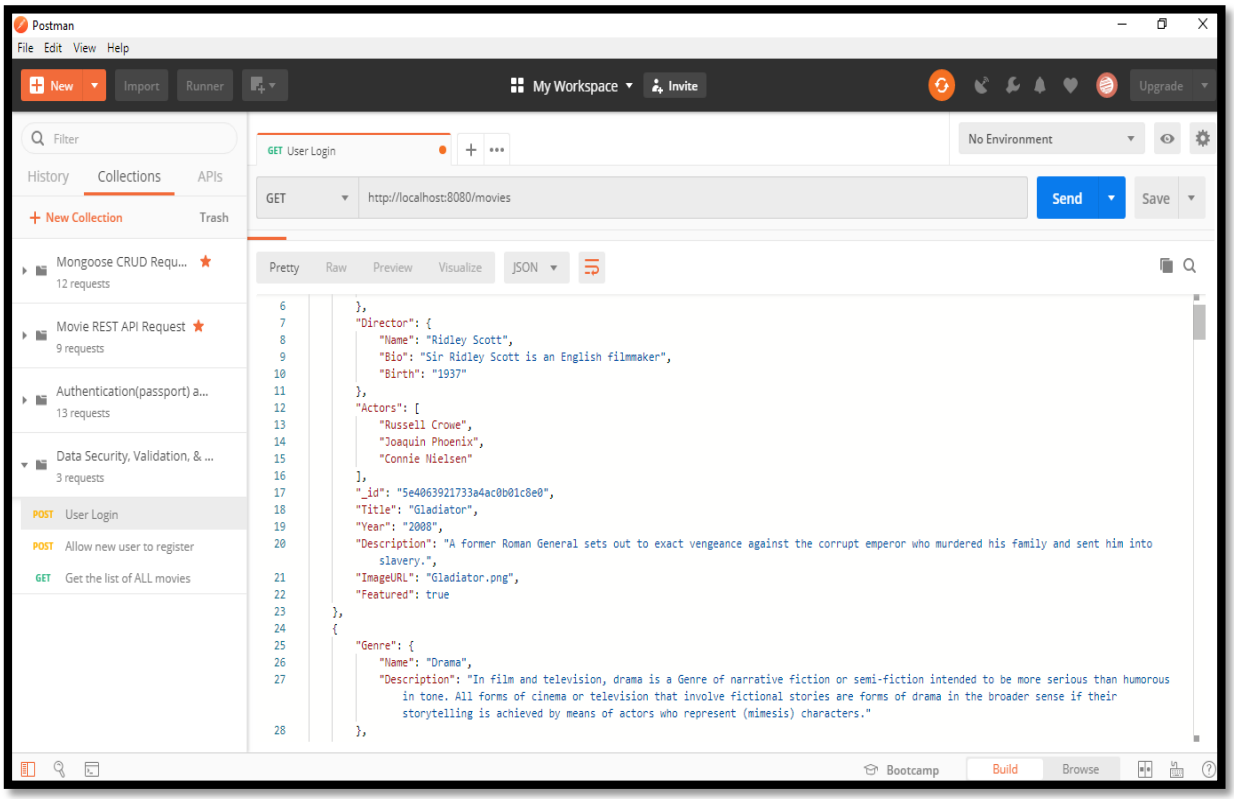


New User login and authorization token

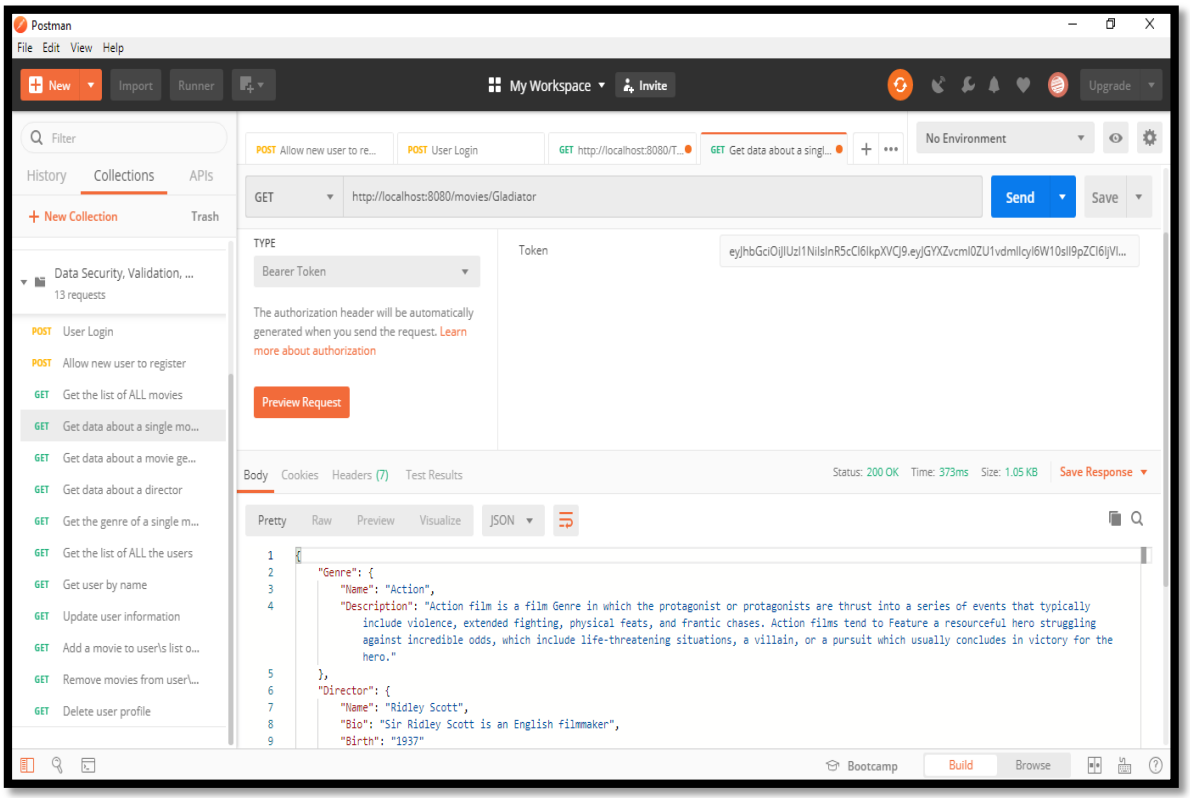


Postman screenshots with authorization token (11 requests)

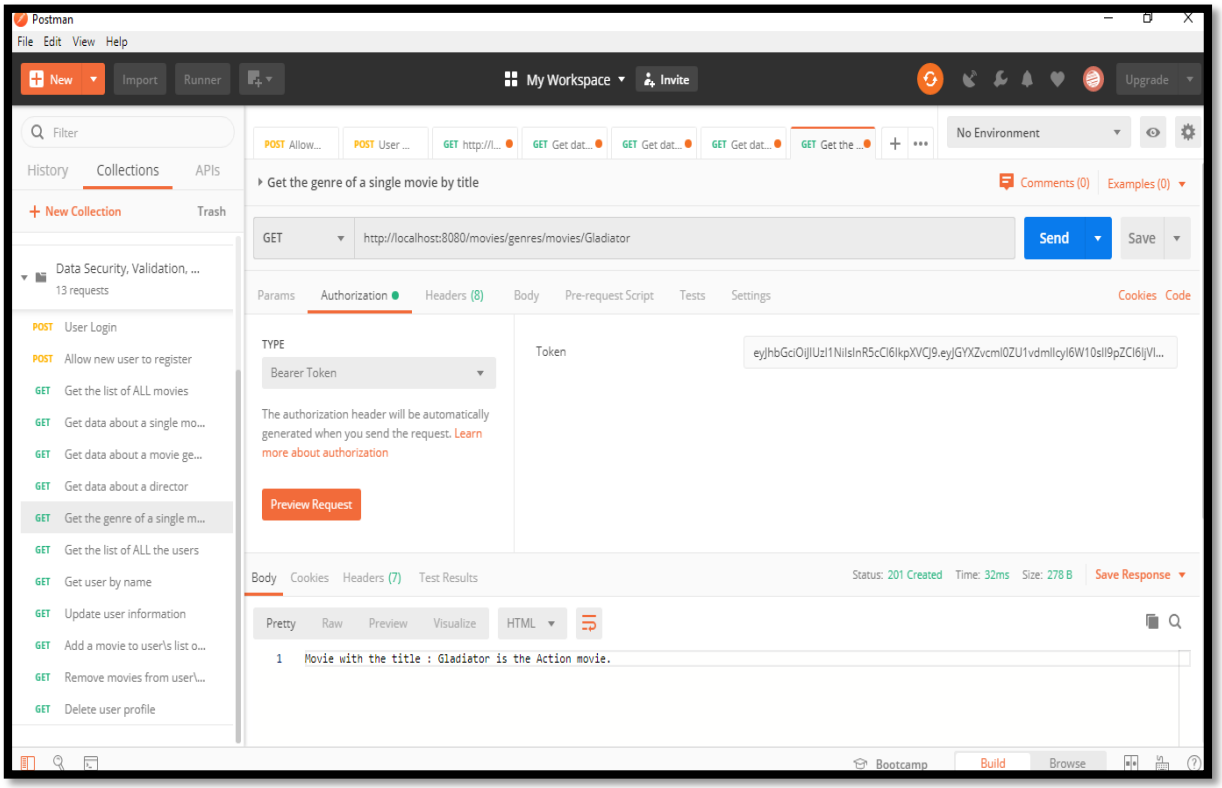
One: Return a list of ALL movies to the user:



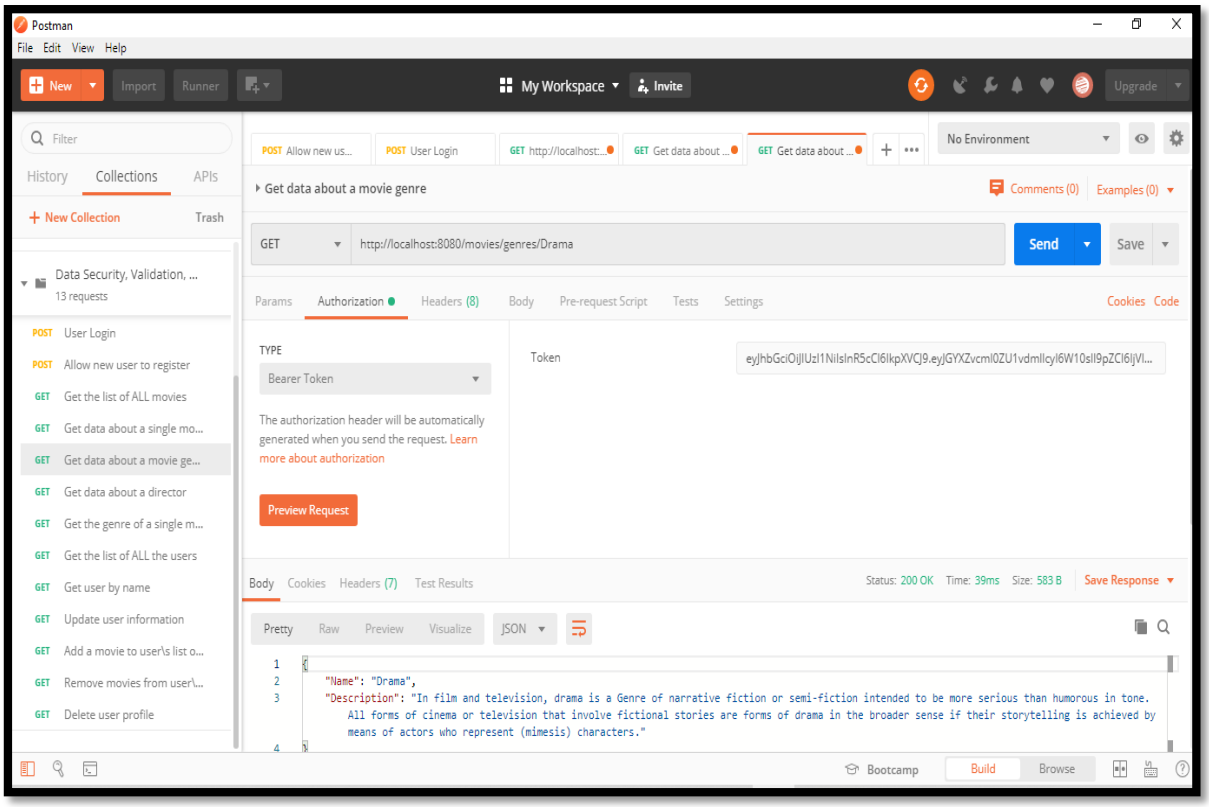
Two: Return data (description, genre, director, image URL, whether it's featured or not) about a single movie by title to the user:



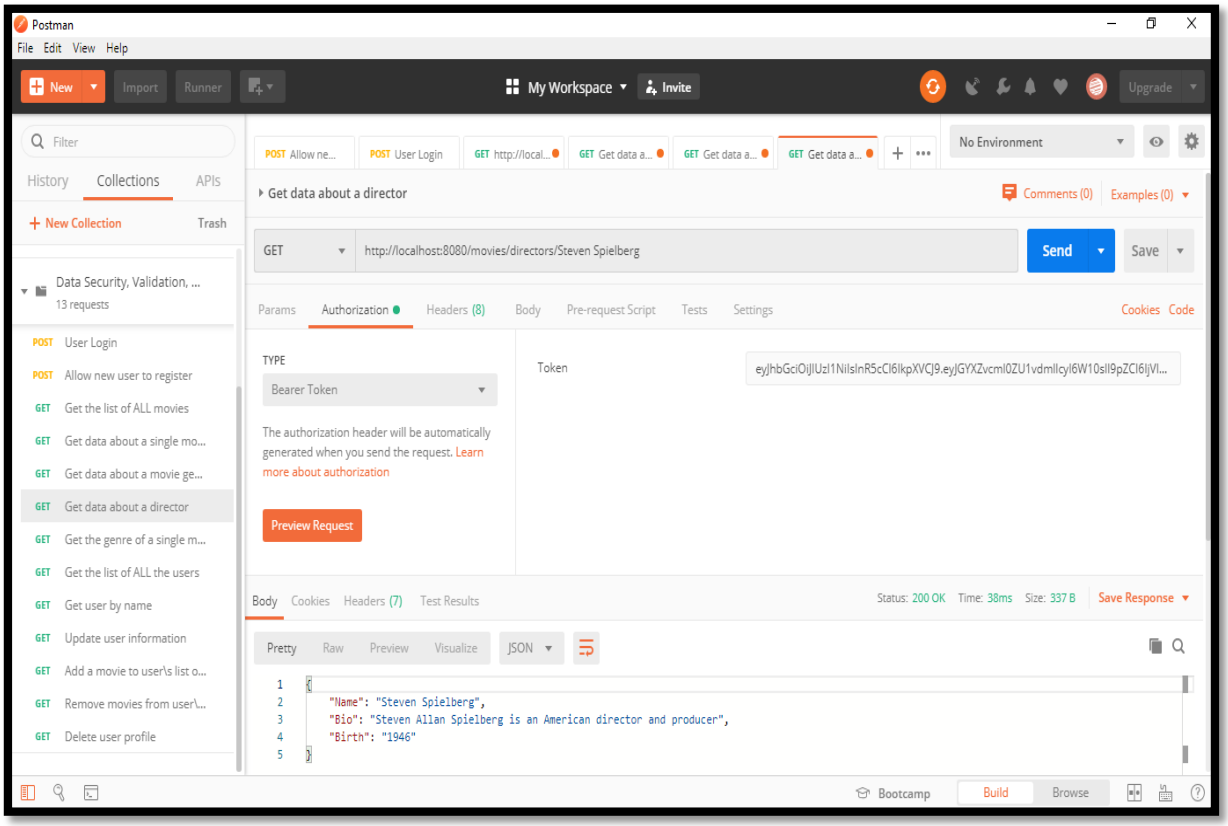
Three: Return data about a genre by the movie title (e.g., "Gladiator"):



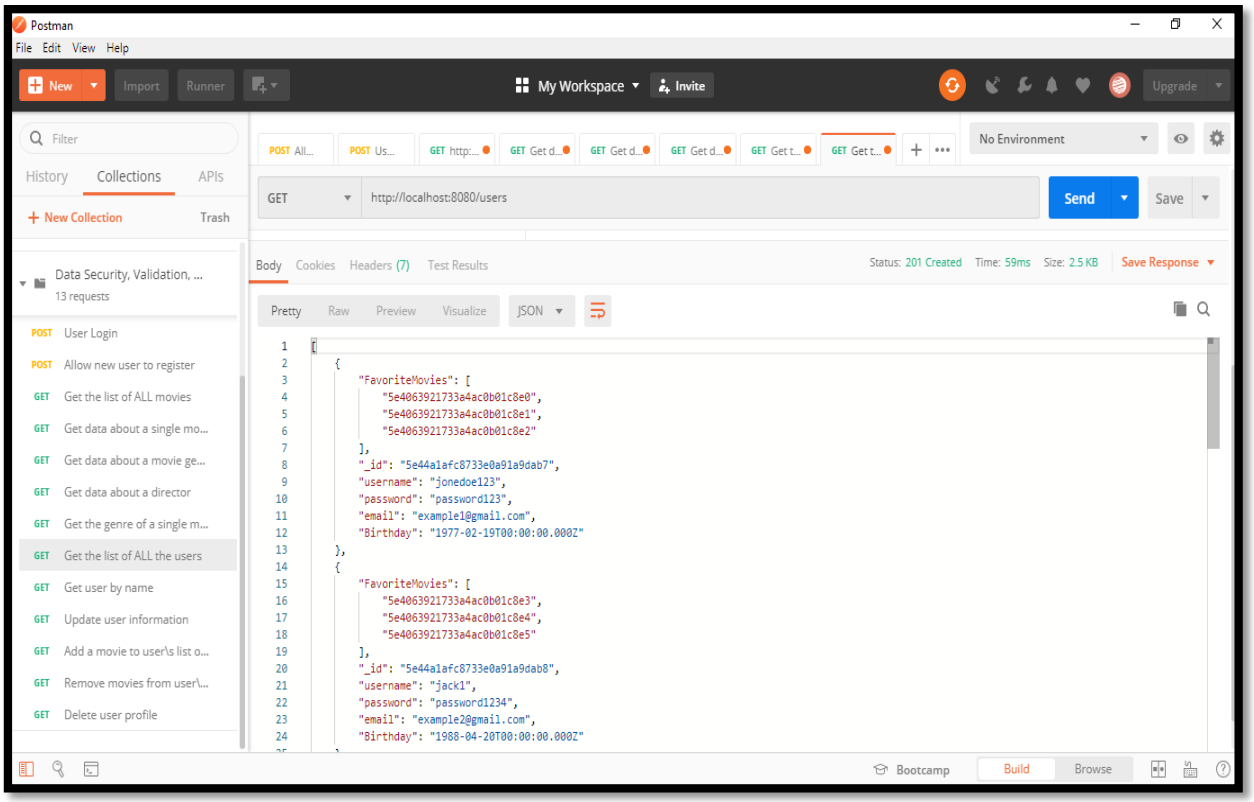
Four: Return data about a genre (description) by the genre Name (e.g., “Drama”):



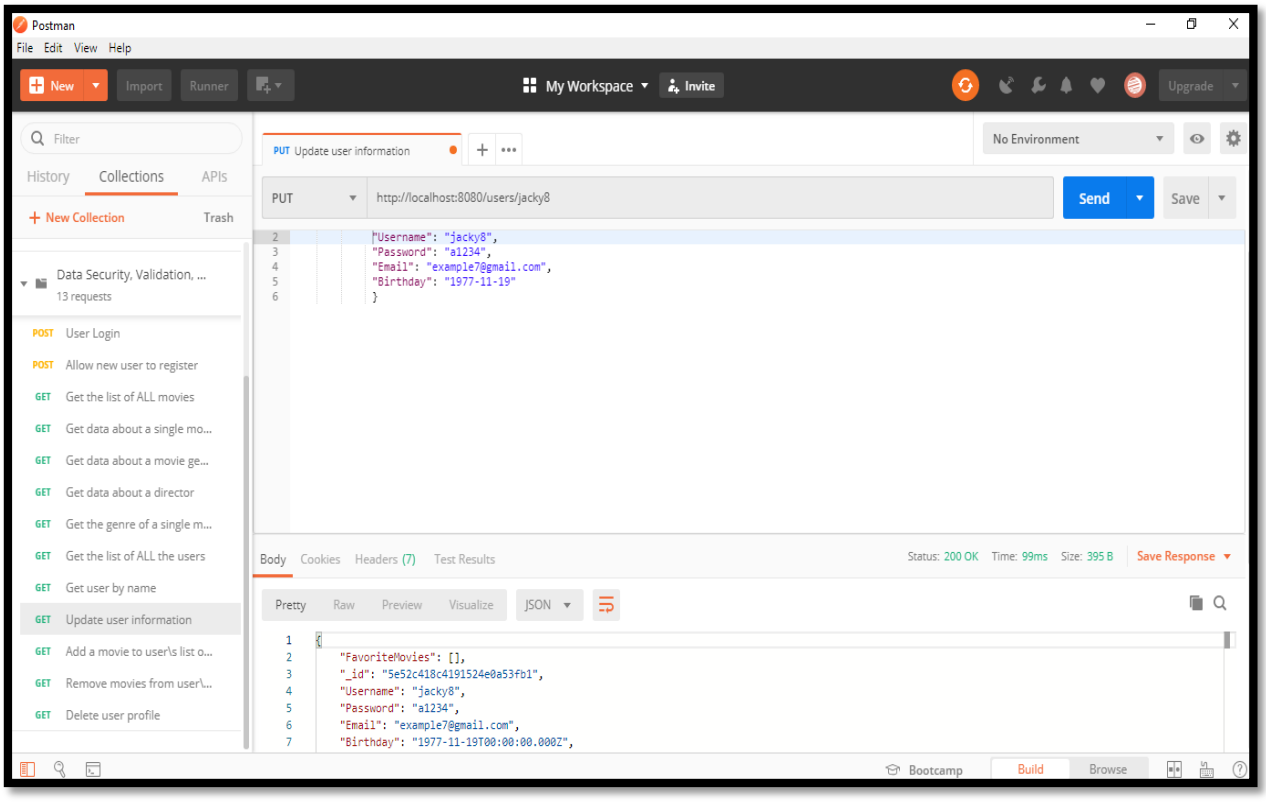
Five: Return data about a director (bio, birth year, death year) by name:



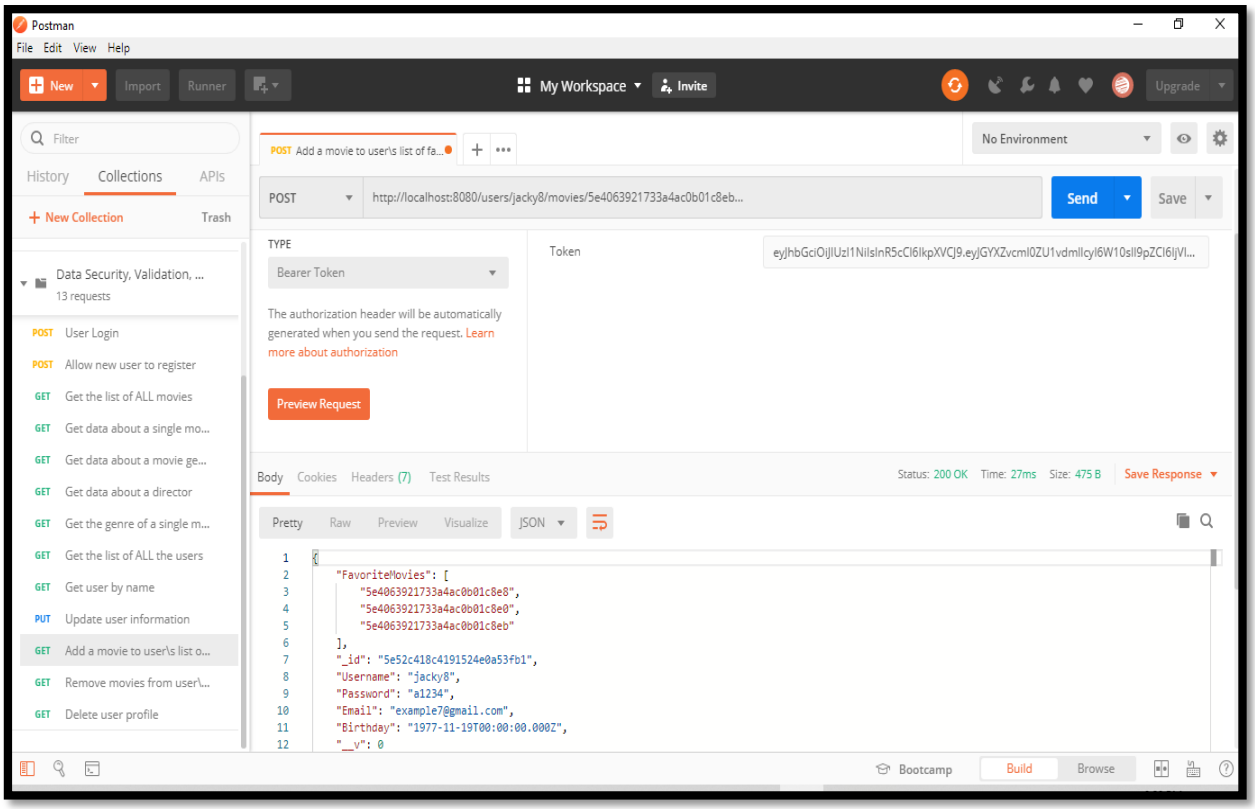
Six: Return a list of ALL users:



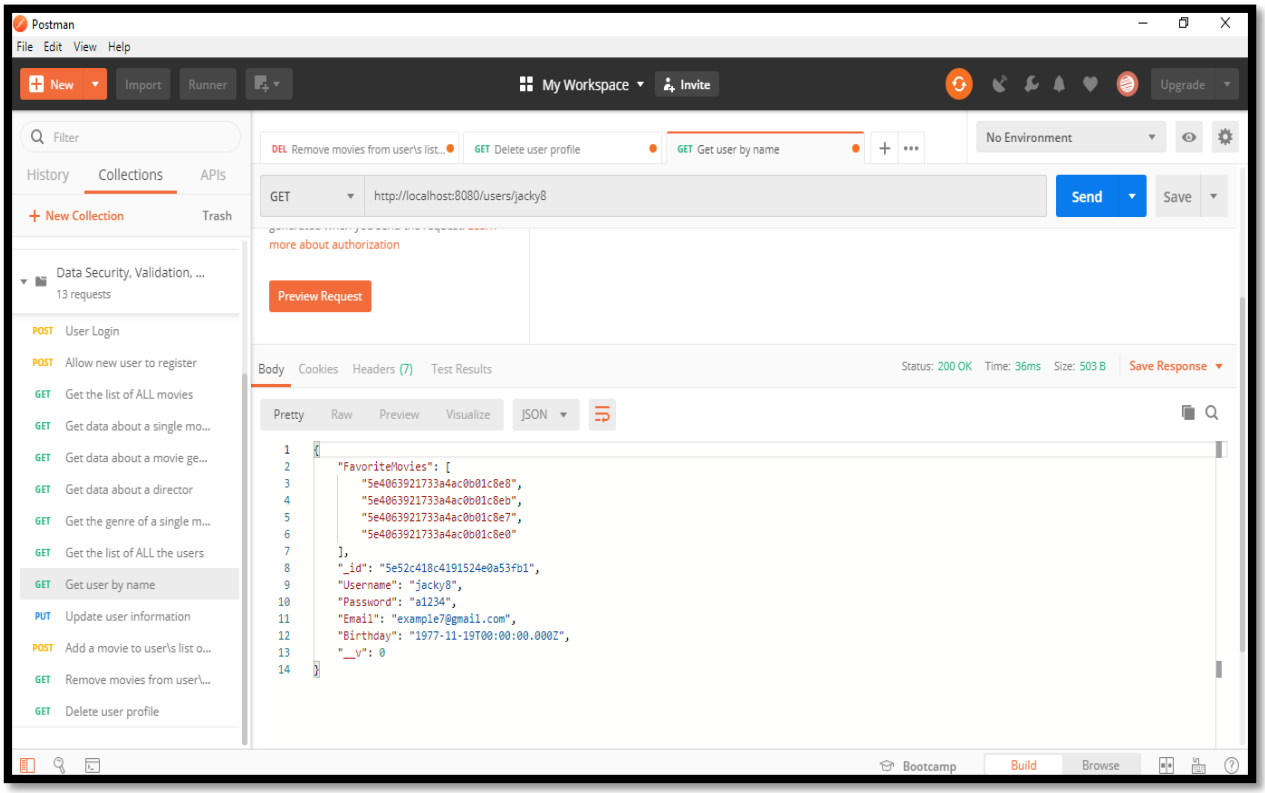
Seven: Allow users to update their user info (username, password, email, date of birth):



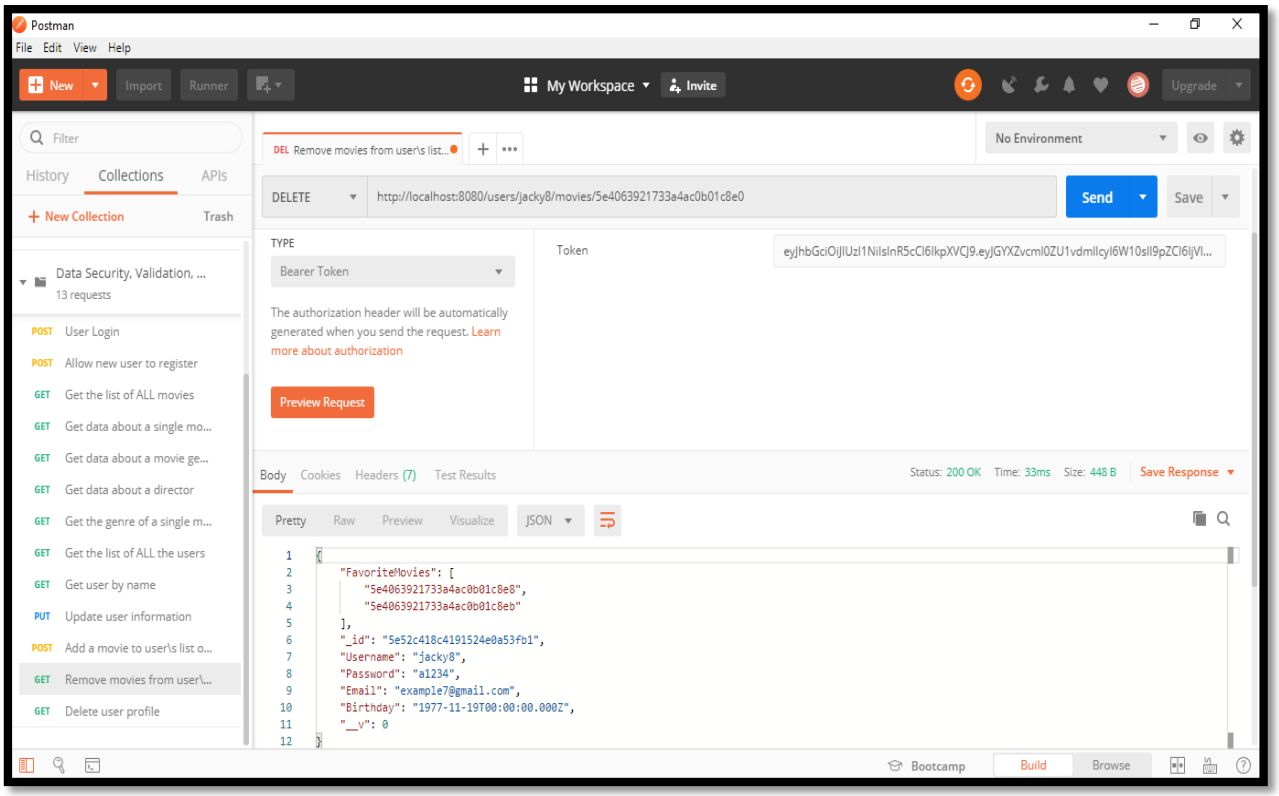
Eight: Allow users to add a movie to their list of favorites:



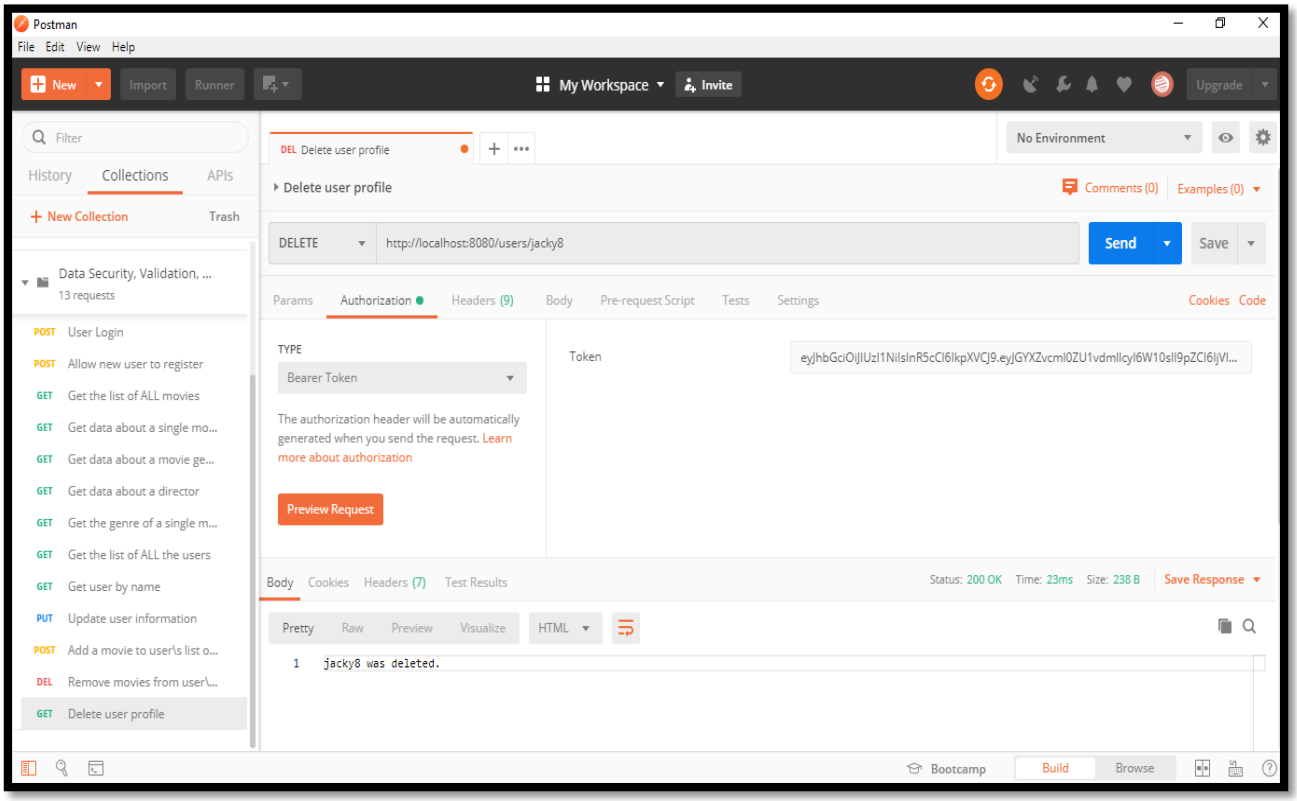
Nine: Get user by Name:



Ten: Allow users to remove a movie from their list of favorites:



Eleven: Allow existing users to deregister with authorization token:



Exercises 2.10b

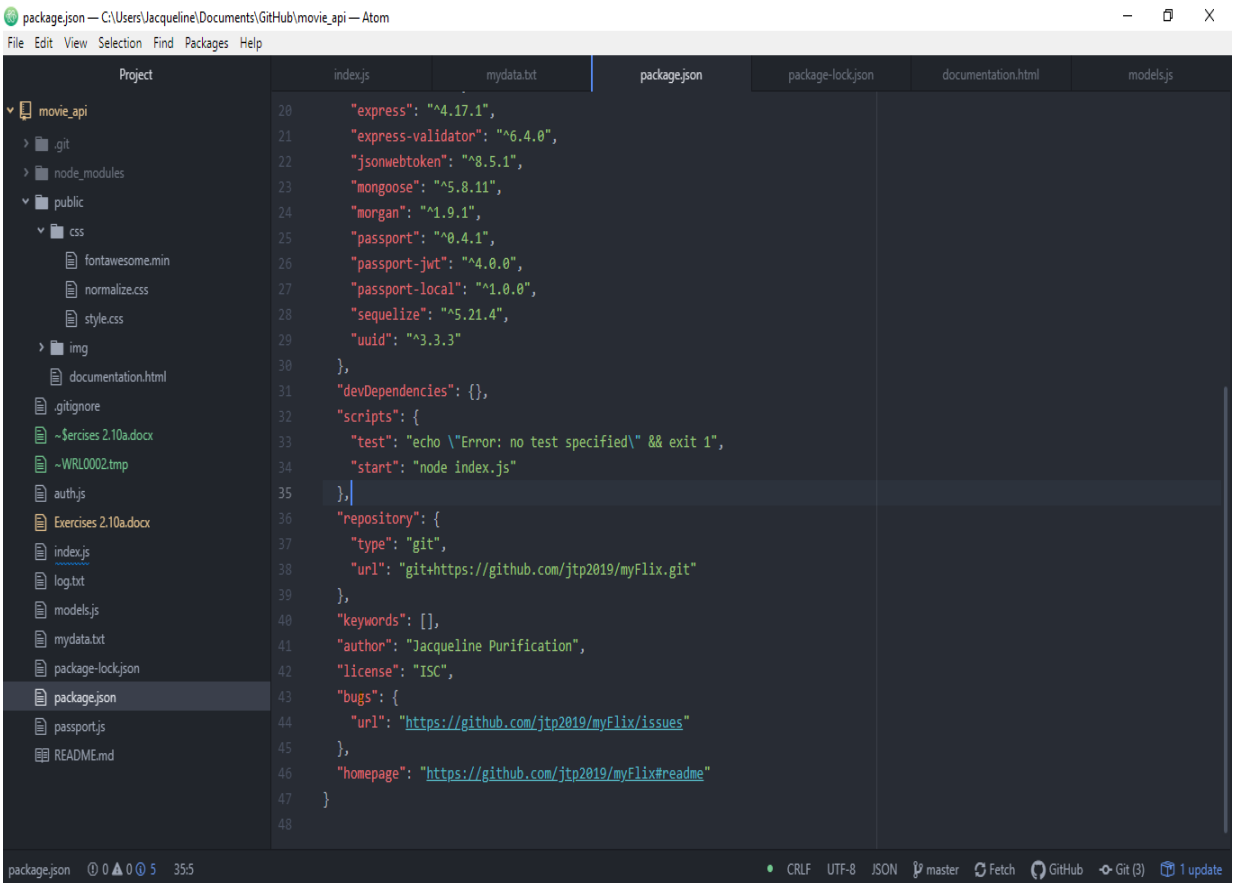
HEROKU

Installed Heroku via command prompt:

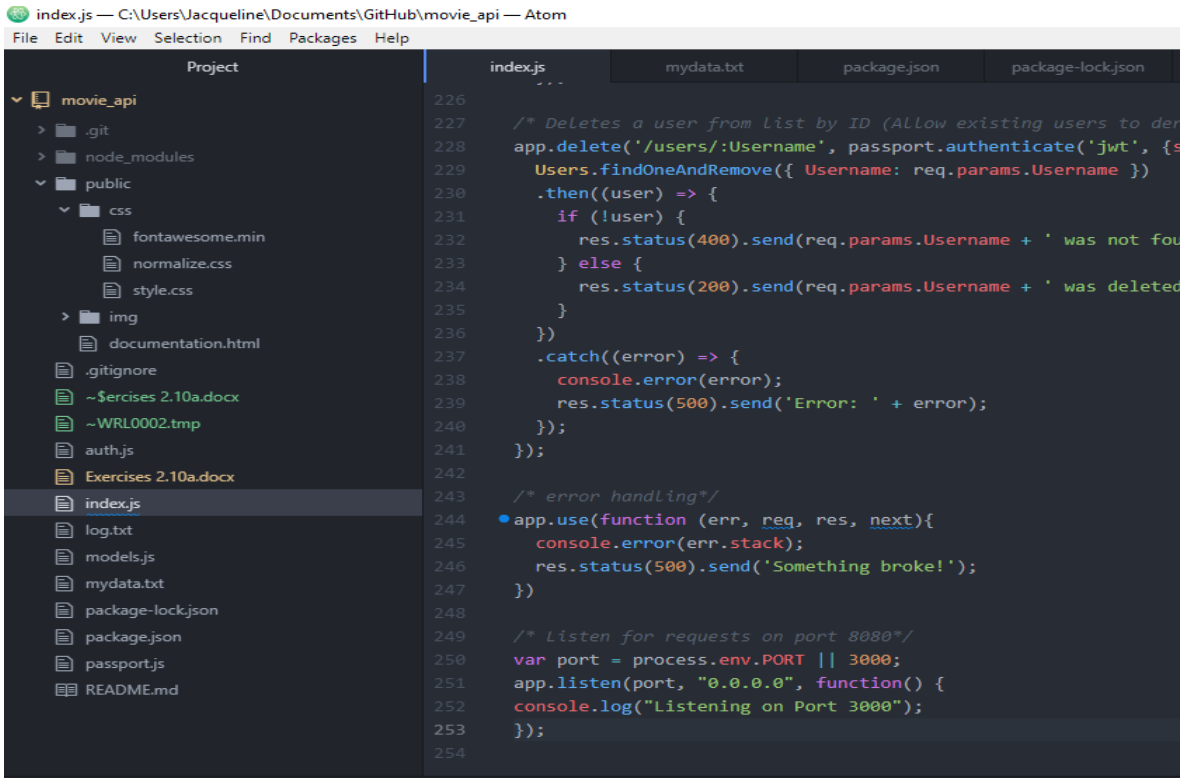
Select Command Prompt - heroku login -i

```
Microsoft Windows [Version 10.0.18362.175]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Jacqueline>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/a98993cf-9fab-409d-b136-2c18c68258dc
Logging in... done
Logged in as jackpr77@gmail.com

C:\Users\Jacqueline>heroku --version
heroku/7.38.2 win32-x86 node-v12.13.0

C:\Users\Jacqueline>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/3a2f0d95-a70d-424f-b9f4-0b3847b57e68
Logging in... done
Logged in as jackpr77@gmail.com
```

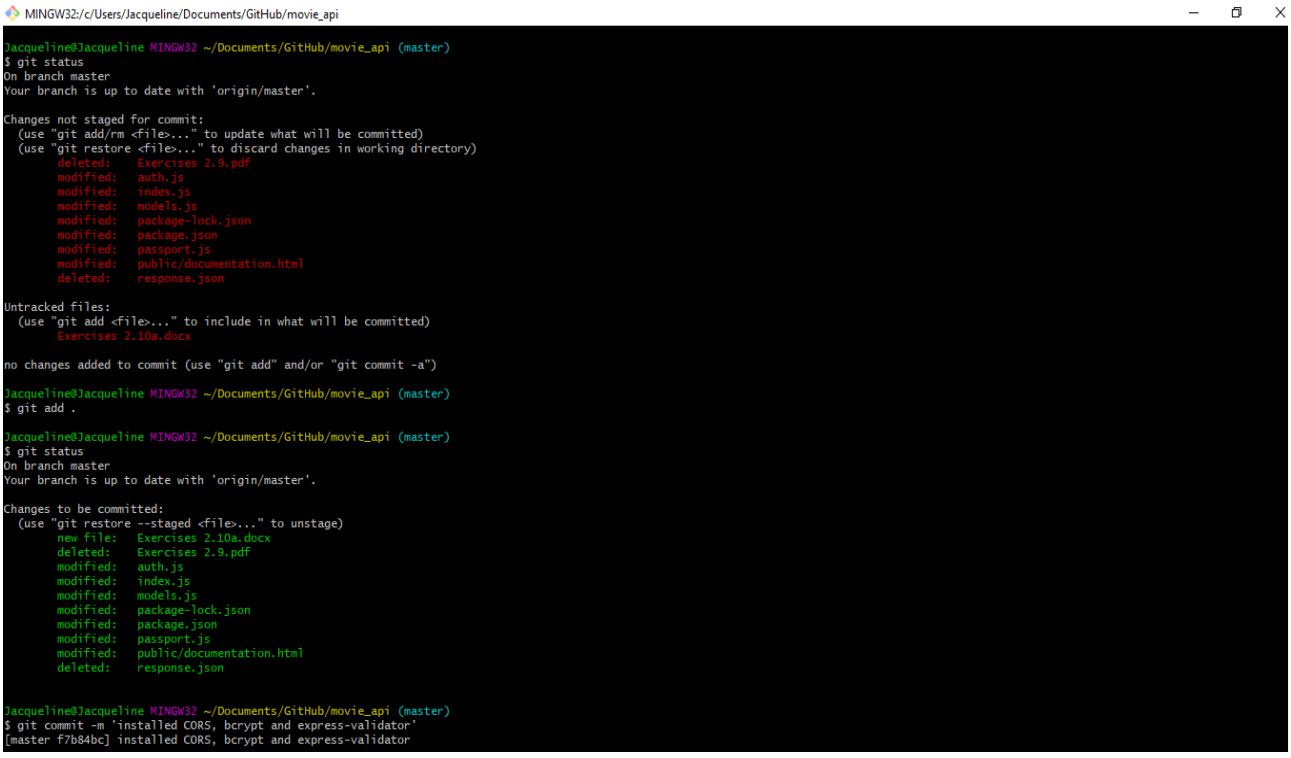
Updated package.json added "start": "node index.js" :



Updated app.listen();



Push changes to Git and heroku master:



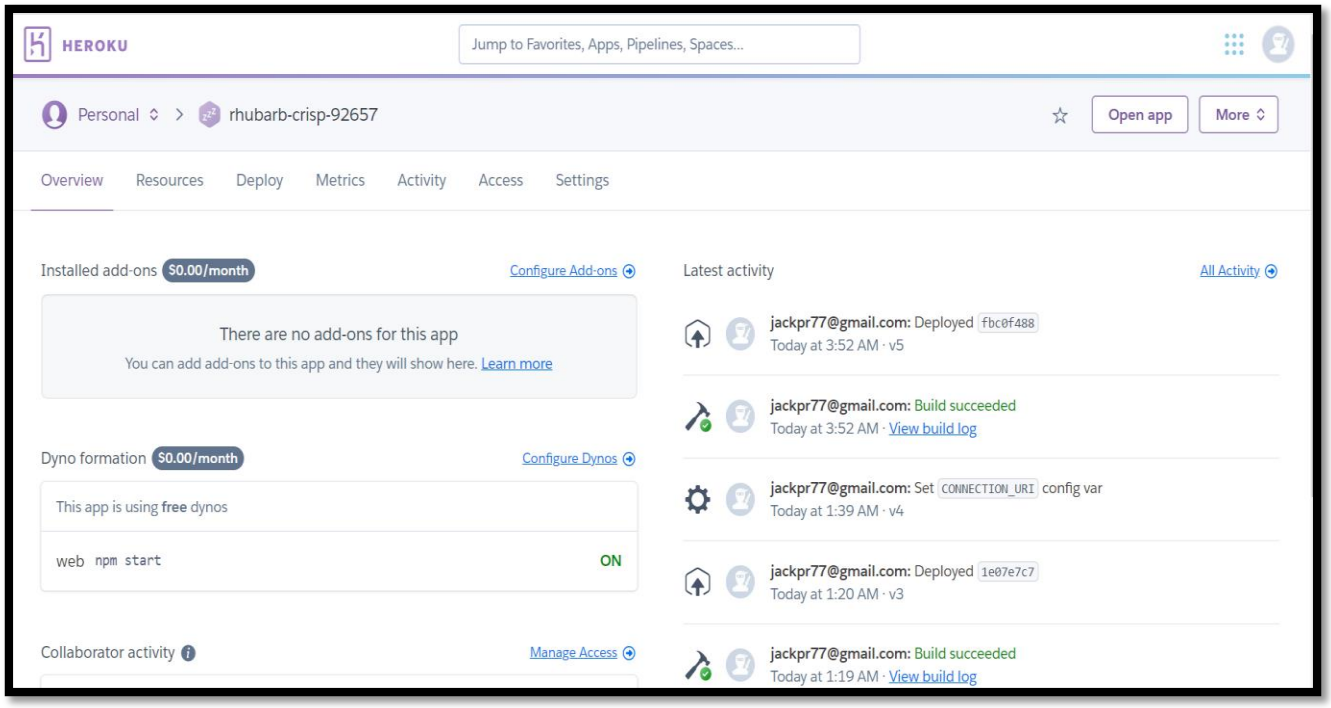
Create and deploy my app on Heroku:

MINGW32:/c/Users/Jacqueline/Documents/GitHub/movie_api

```
Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ git push heroku
Enumerating objects: 109, done.
Counting objects: 100% (109/109), done.
Delta compression using up to 4 threads
Compressing objects: 100% (105/105), done.
Writing objects: 100% (109/109), 8.93 MiB | 91.00 KiB/s, done.
Total 109 (delta 39), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Node.js app detected
remote:
remote: -----> Creating runtime environment
remote:
remote:        NPM_CONFIG_LOGLEVEL=error
remote:        NODE_ENV=production
remote:        NODE_MODULES_CACHE=true
remote:        NODE_VERBOSE=false
remote:
remote: -----> Installing binaries
remote:        engines.node (package.json):  unspecified
remote:        engines.npm (package.json):   unspecified (use default)
remote:
remote:        Resolving node version 12.x...
remote:        Downloading and installing node 12.16.1...
remote:        Using default npm version: 6.13.4
remote:
remote: -----> Installing dependencies
remote:        Installing node modules (package.json + package-lock)
remote:
remote:        > bcrypt@4.0.0 install /tmp/build_52972d4e85379955ede1926c4b87e57a/node_modules/bcrypt
remote:        > node-pre-gyp install --fallback-to-build
remote:
remote:        [bcrypt] Success: "/tmp/build_52972d4e85379955ede1926c4b87e57a/node_modules/bcrypt/lib/binding/napi-v3/bcrypt_lib.node" is installed via remote
remote:        added 196 packages from 341 contributors and audited 528 packages in 6.086s
remote:
remote:        2 packages are looking for funding
remote:          run `npm fund` for details
remote:
remote:        found 0 vulnerabilities
remote:
remote:
remote: -----> Build
remote:
remote: -----> Caching build
remote:        - node_modules
remote:
remote: -----> Pruning devDependencies
```

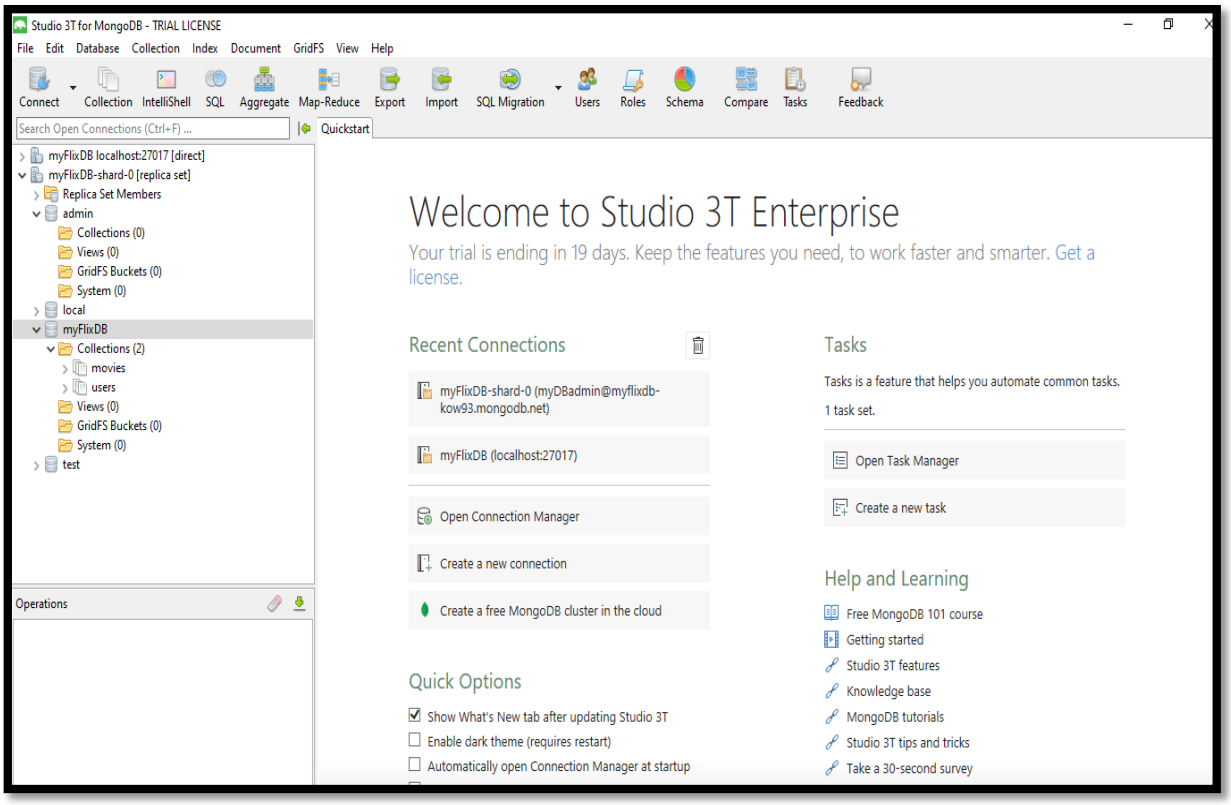
MINGW32:/c/Users/Jacqueline/Documents/GitHub/movie_api

```
remote:        Resolving node version 12.x...
remote:        Downloading and installing node 12.16.1...
remote:        Using default npm version: 6.13.4
remote:
remote: -----> Installing dependencies
remote:        Installing node modules (package.json + package-lock)
remote:
remote:        > bcrypt@4.0.0 install /tmp/build_52972d4e85379955ede1926c4b87e57a/node_modules/bcrypt
remote:        > node-pre-gyp install --fallback-to-build
remote:
remote:        [bcrypt] Success: "/tmp/build_52972d4e85379955ede1926c4b87e57a/node_modules/bcrypt/lib/binding/napi-v3/bcrypt_lib.node" is installed via remote
remote:        added 196 packages from 341 contributors and audited 528 packages in 6.086s
remote:
remote:        2 packages are looking for funding
remote:          run `npm fund` for details
remote:
remote:        found 0 vulnerabilities
remote:
remote:
remote: -----> Build
remote:
remote: -----> Caching build
remote:        - node_modules
remote:
remote: -----> Pruning devDependencies
remote:        audited 528 packages in 1.73s
remote:
remote:        2 packages are looking for funding
remote:          run `npm fund` for details
remote:
remote:        found 0 vulnerabilities
remote:
remote:
remote: -----> Build succeeded!
remote: -----> Discovering process types
remote:        Procfile declares types     -> (none)
remote:        Default types for buildpack -> web
remote:
remote: -----> Compressing...
remote:        Done: 28.7M
remote: -----> Launching...
remote:        Released v7
remote:        https://myflix-db1.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/myflix-db1.git
 * [new branch]      master -> master

Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
```

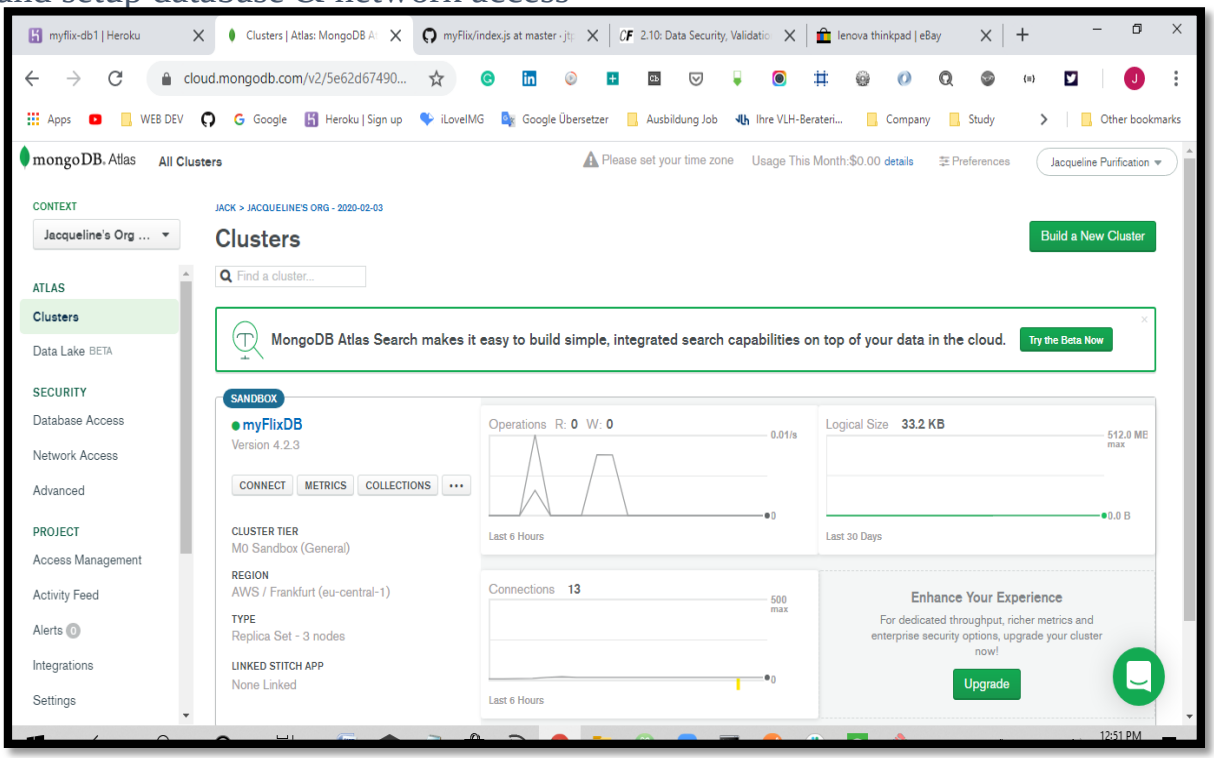


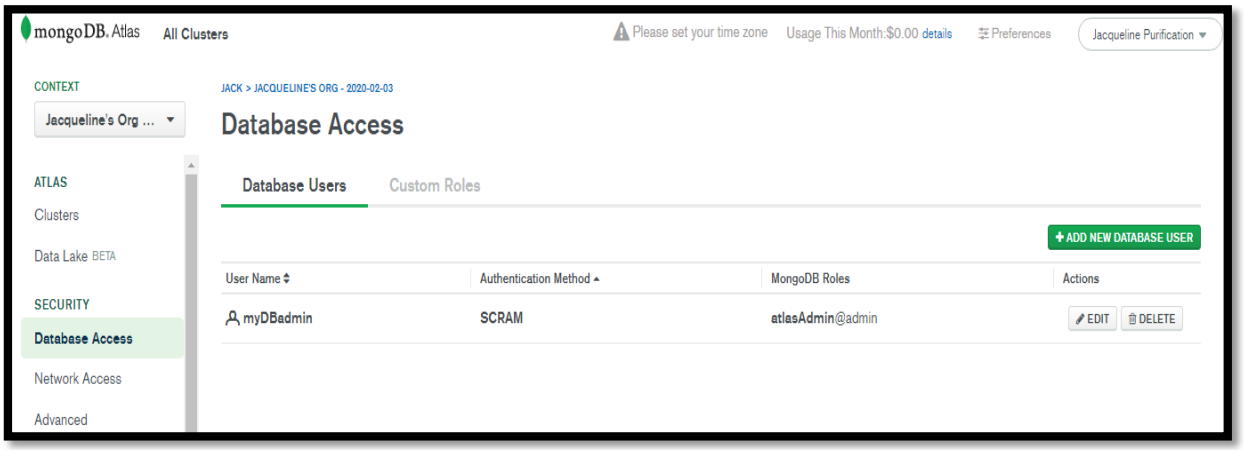
Upload database to MongoDB Atlas via Studio 3T

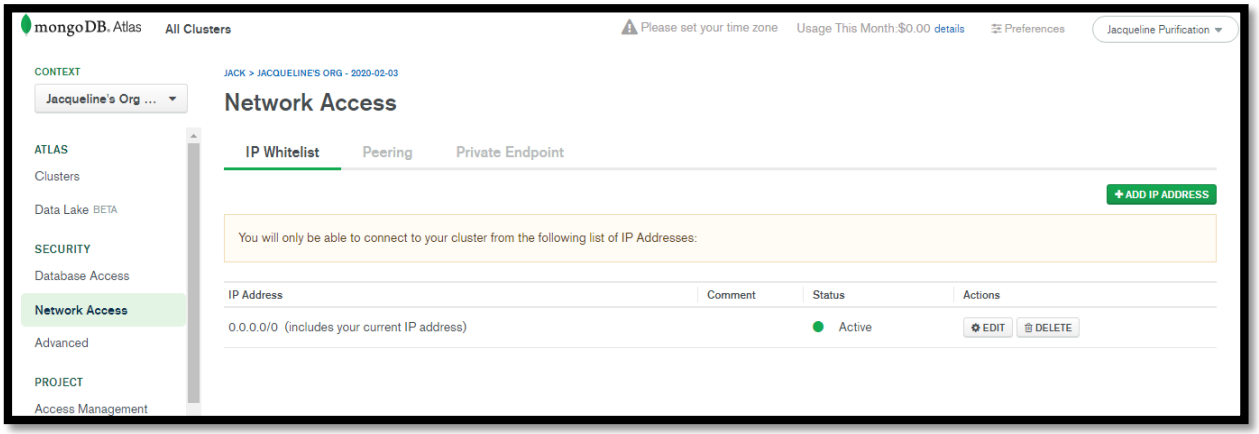
1st Step: Installed Studio 3T for 32 bit Win 10, Connect S3T with mongodb database (device) and added all collections to myFlixDB (localhost 27027) folder. After successfully getting the collections export & execute those data in Json format to my PC. Which will automatic created a folder in my PC named myFlixDB.



2nd Step: Open account on MongoDB atlas, created a cluster named myFlixDB, and setup database & network access

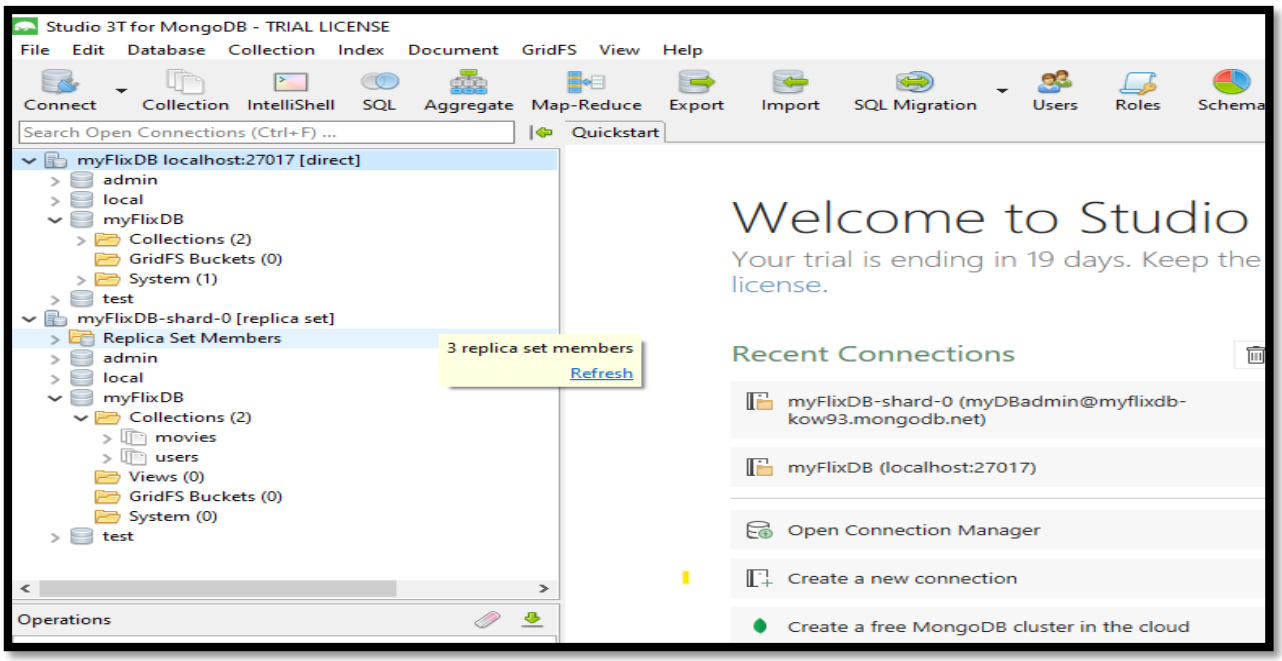




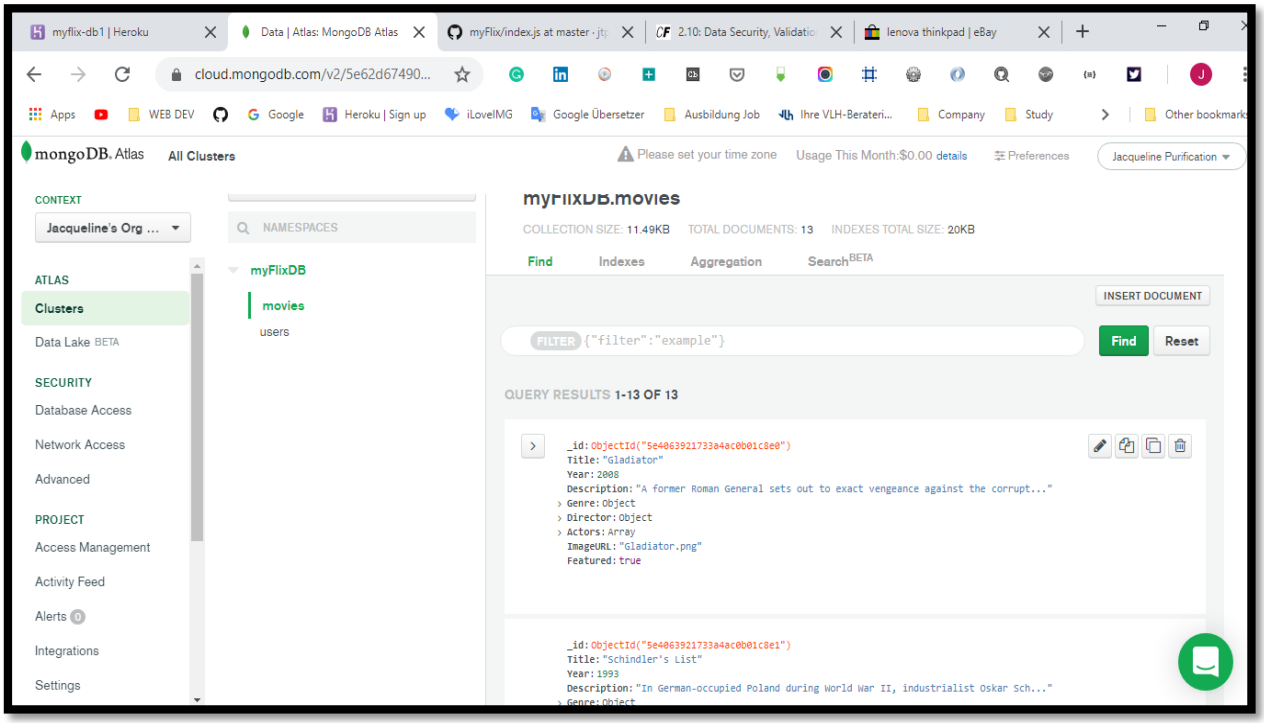


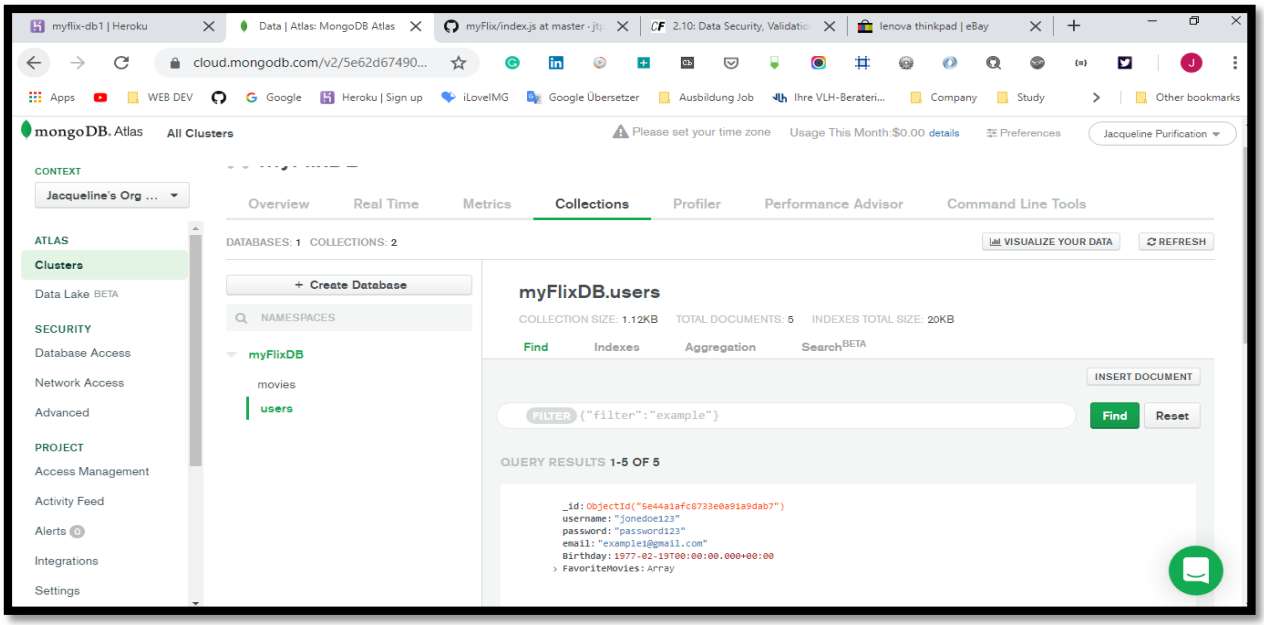
3rd Step: Connected Studio 3T with MonngoDB Cluster

Open a new connection named myFlixDB-shard-0 and export & executed all data in Json format from myFlixDB (localhost 27027) to my PC in different folder named myFlixDB and from that folder import data to the myFlixDB-shard-0 folder.



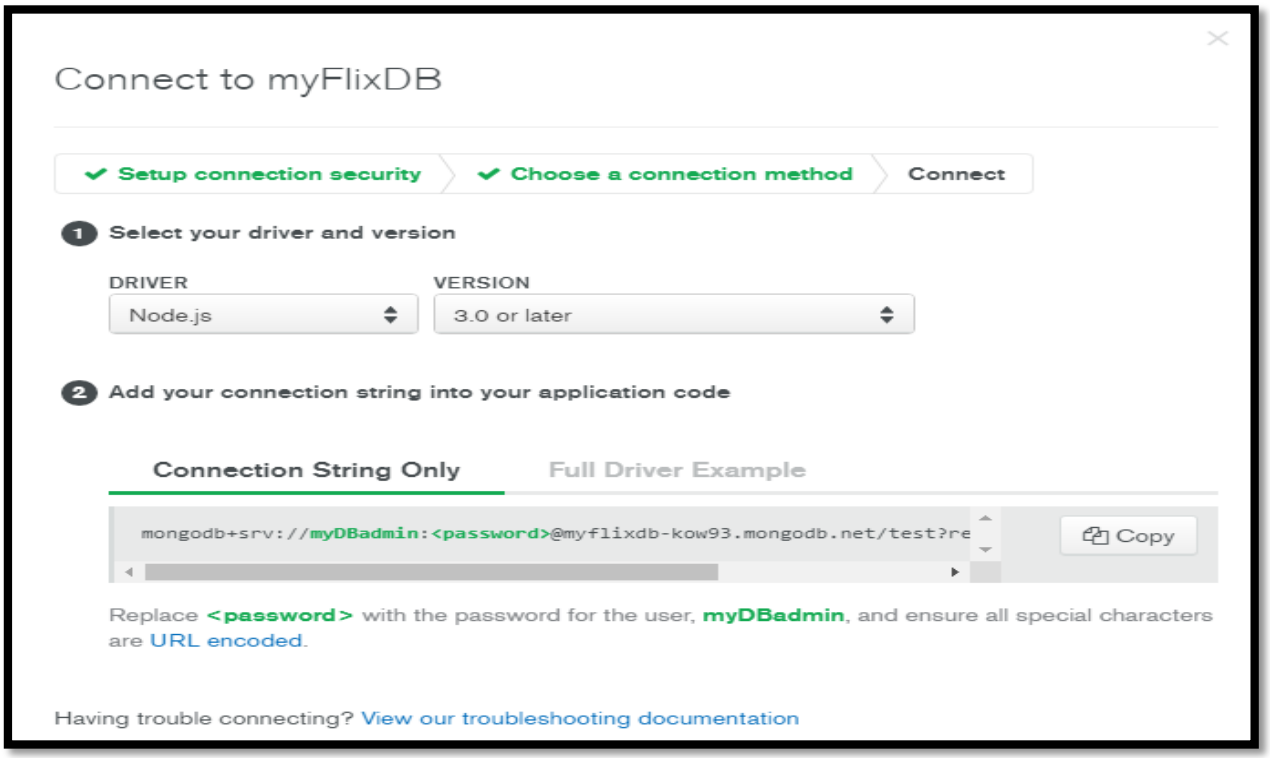
Now the collections are showing on the MongoDB Atlas

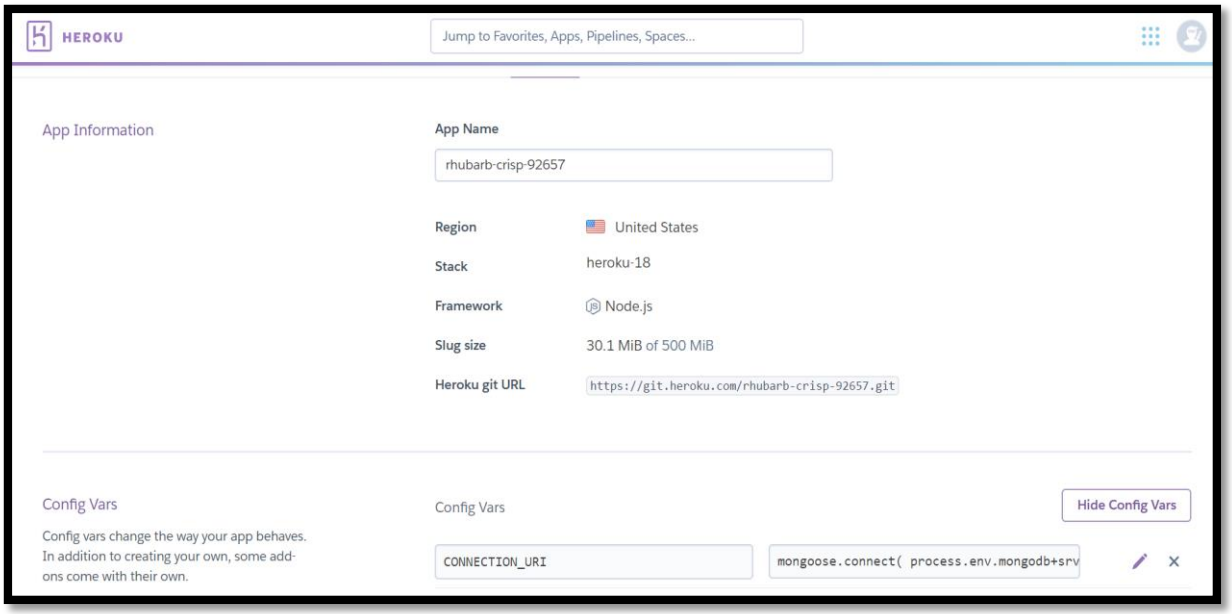




Connected MongoDB atlas with Heroku (GitHub)

1st Step: From MongoDB atlas collected connection string 'mongodb+srv://myDBadmin:12345@myflixdb-kow93.mongodb.net/test?retryWrites=true&w=majority' changed the format to mongoose.connect(process.env.mongodb+srv://myDBadmin:12345@myflixdb-kow93.mongodb.net/myFlixDB?retryWrites=true&w=majority', { useNewUrlParser: true, useUnifiedTopology: true }); and added that URL to index.js and paste it into Heroku app Config Vars.





2nd Step: Push all changes to Heroku and GitHub master.

```
MINGW32:/c/Users/Jacqueline/Documents/GitHub/movie_api
Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ git add .

Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ git commit -am " Correction port connection"
[master ec1eb00] Correction port connection
 1 file changed, 1 insertion(+), 1 deletion(-)

Jacqueline@Jacqueline MINGW32 ~/Documents/GitHub/movie_api (master)
$ git push heroku master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 309 bytes | 103.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Node.js app detected
remote:
remote: -----> Creating runtime environment
remote:
remote:        NPM_CONFIG_LOGLEVEL=error
remote:        NODE_ENV=production
remote:        NODE_MODULES_CACHE=true
remote:        NODE_VERBOSE=false
remote:
remote: -----> Installing binaries
remote:        engines.node (package.json):  unspecified
remote:        engines.npm (package.json):   unspecified (use default)
remote:
remote:        Resolving node version 12.x...
remote:        Downloading and installing node 12.16.1...
remote:        Using default npm version: 6.13.4
remote:
remote: -----> Restoring cache
remote:        - node_modules
remote:
remote: -----> Installing dependencies
remote:        Installing node modules (package.json + package-lock)
remote:        audited 528 packages in 1.667s
remote:
remote:        2 packages are looking for funding
remote:          run `npm fund` for details
remote:
remote:        found 0 vulnerabilities
remote:
remote:
```

New URL from Heroku

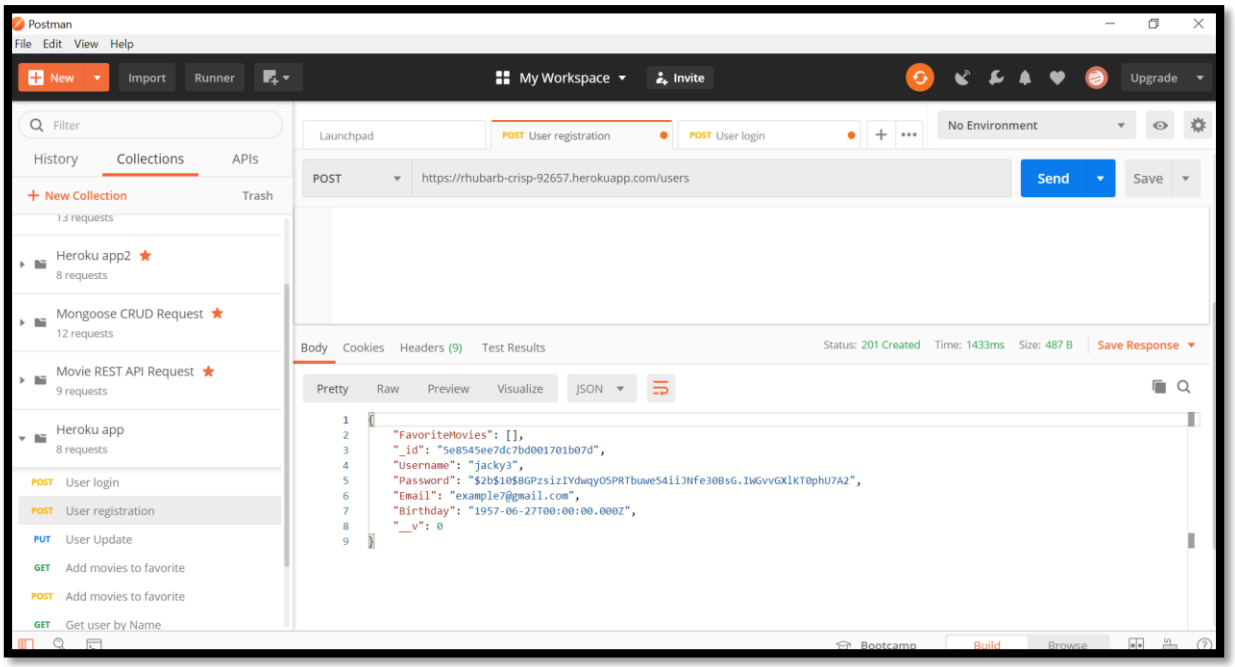
https://rhubarb-crisp-92657.herokuapp.com/

3rd Step: Checked Heroku logs

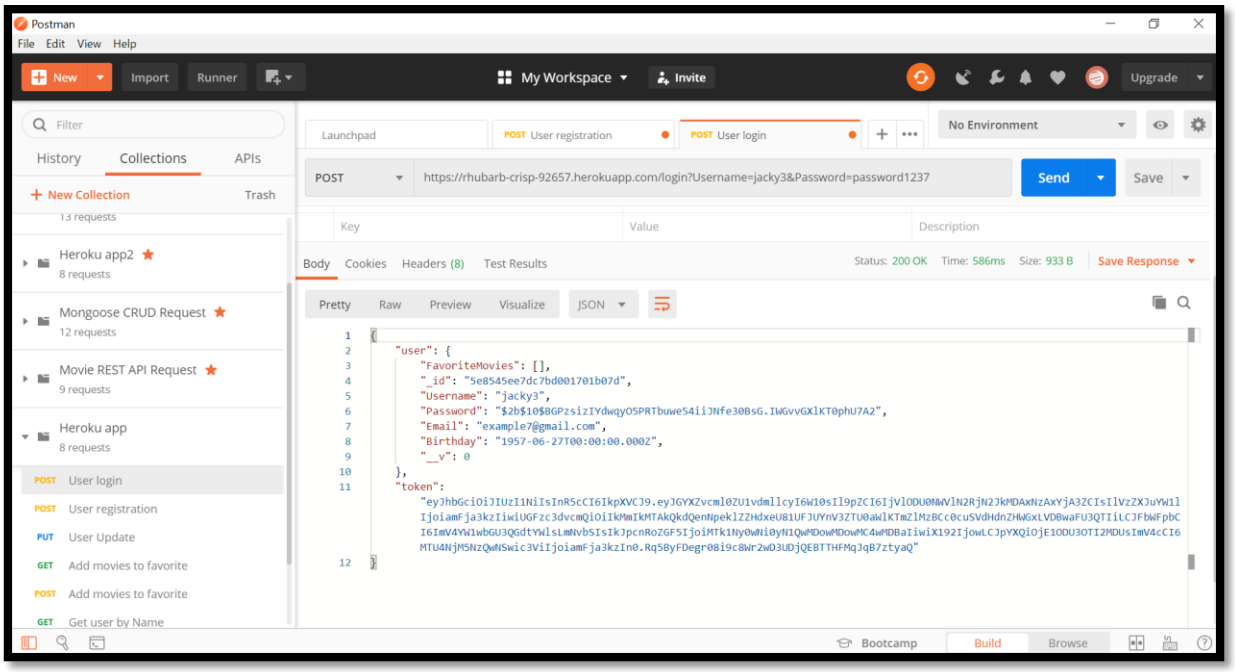
```
Admin@Jack MINGW64 ~/Documents/GitHub/movie_api (master)
$ heroku logs --app rhubarb-crisp-92657
2020-04-01T23:00:54.914748+00:00 app[api]: Initial release by user jackpr77@gmai
l.com
2020-04-01T23:00:55.297417+00:00 app[api]: Release v2 created by user jackpr77@g
mail.com
2020-04-01T23:00:55.297417+00:00 app[api]: Enable Logplex by user jackpr77@gmail
.com
2020-04-01T23:00:54.914748+00:00 app[api]: Release v1 created by user jackpr77@g
mail.com
2020-04-01T23:19:50.000000+00:00 app[api]: Build started by user jackpr77@gmail.
com
2020-04-01T23:20:11.251937+00:00 app[api]: Deploy 1e07e7c7 by user jackpr77@gmai
l.com
2020-04-01T23:20:11.251937+00:00 app[api]: Release v3 created by user jackpr77@g
mail.com
2020-04-01T23:20:11.271106+00:00 app[api]: Scaled to web@1:Free by user jackpr77
@gmail.com
2020-04-01T23:20:12.000000+00:00 app[api]: Build succeeded
2020-04-01T23:20:16.773156+00:00 app[web.1]:
2020-04-01T23:20:16.773179+00:00 app[web.1]: > movie_api@1.0.0 start /app
2020-04-01T23:20:16.773180+00:00 app[web.1]: > node index.js
2020-04-01T23:20:16.773180+00:00 app[web.1]:
2020-04-01T23:20:17.540293+00:00 app[web.1]: Listening on Port 3000
```

Postman screenshots (testing code before deploying via Heroku and MongoDB Atlas)

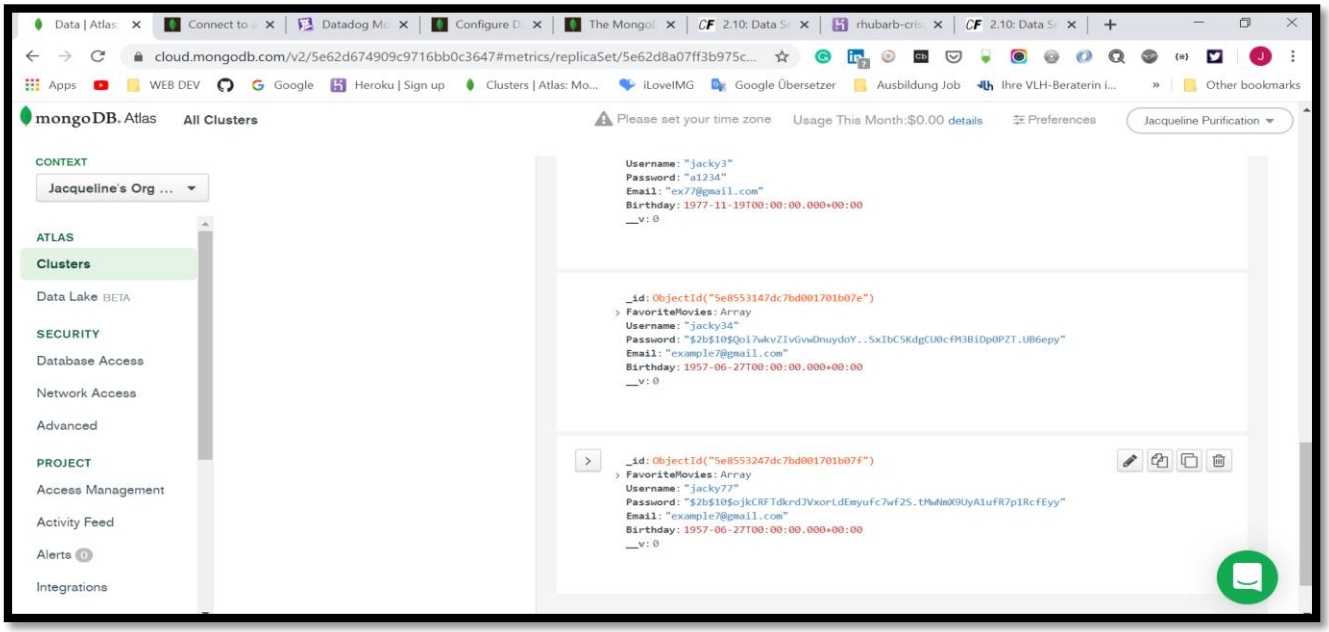
New User Registration



New User login and authorization token

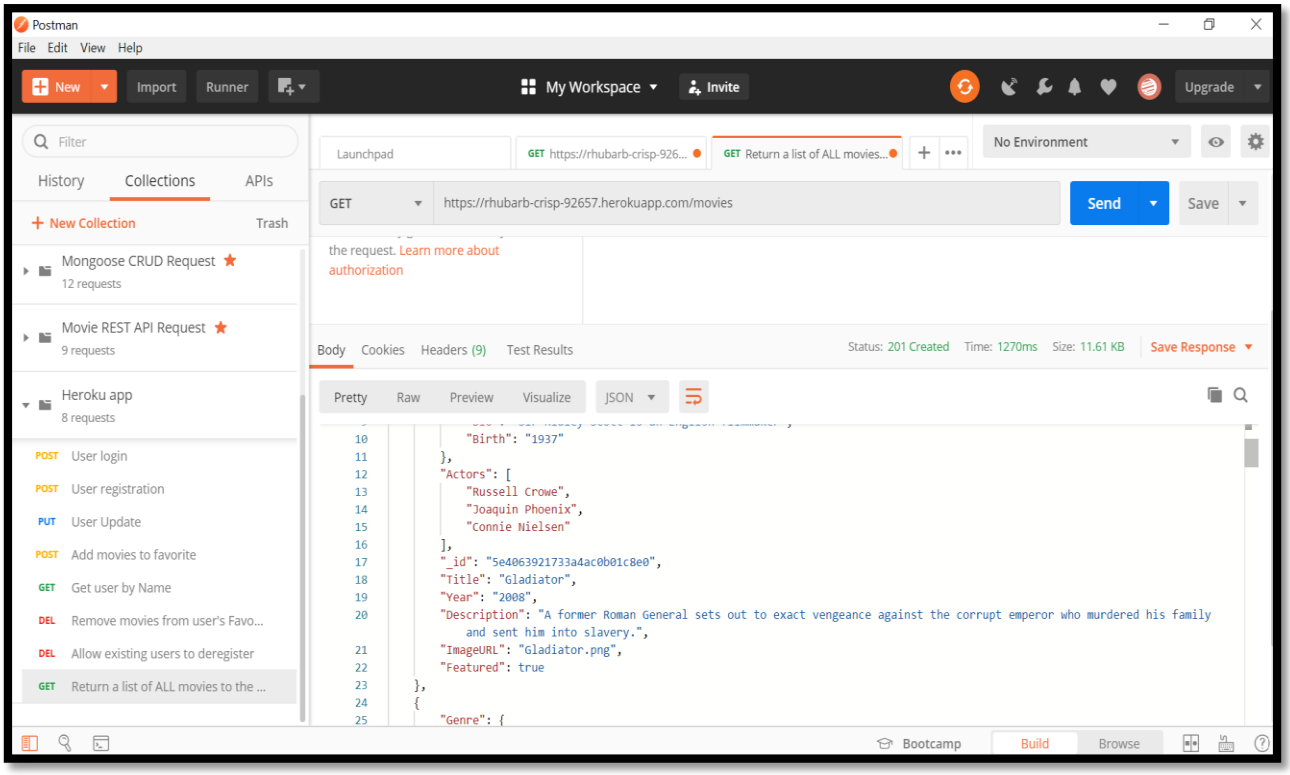


New users are successfully added with the MongoDB Atlas

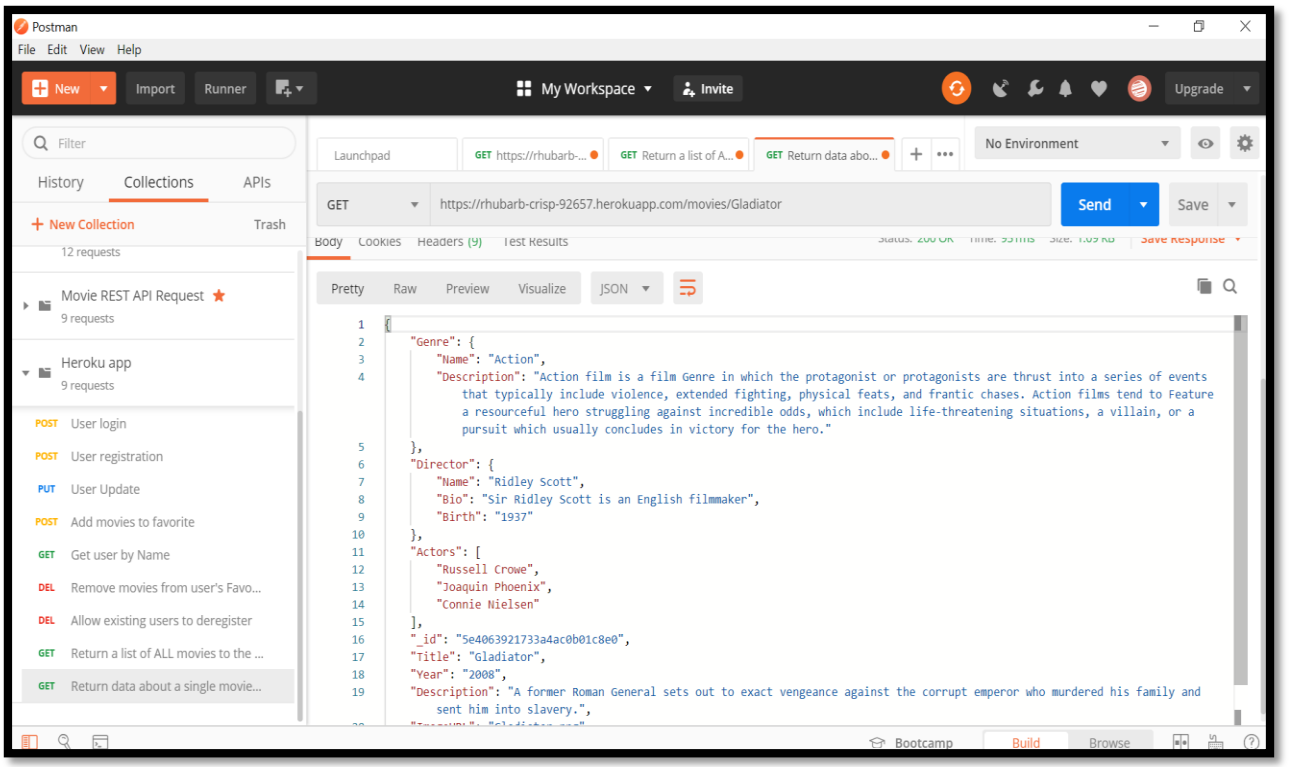


Postman screenshots with authorization token (11 requests)

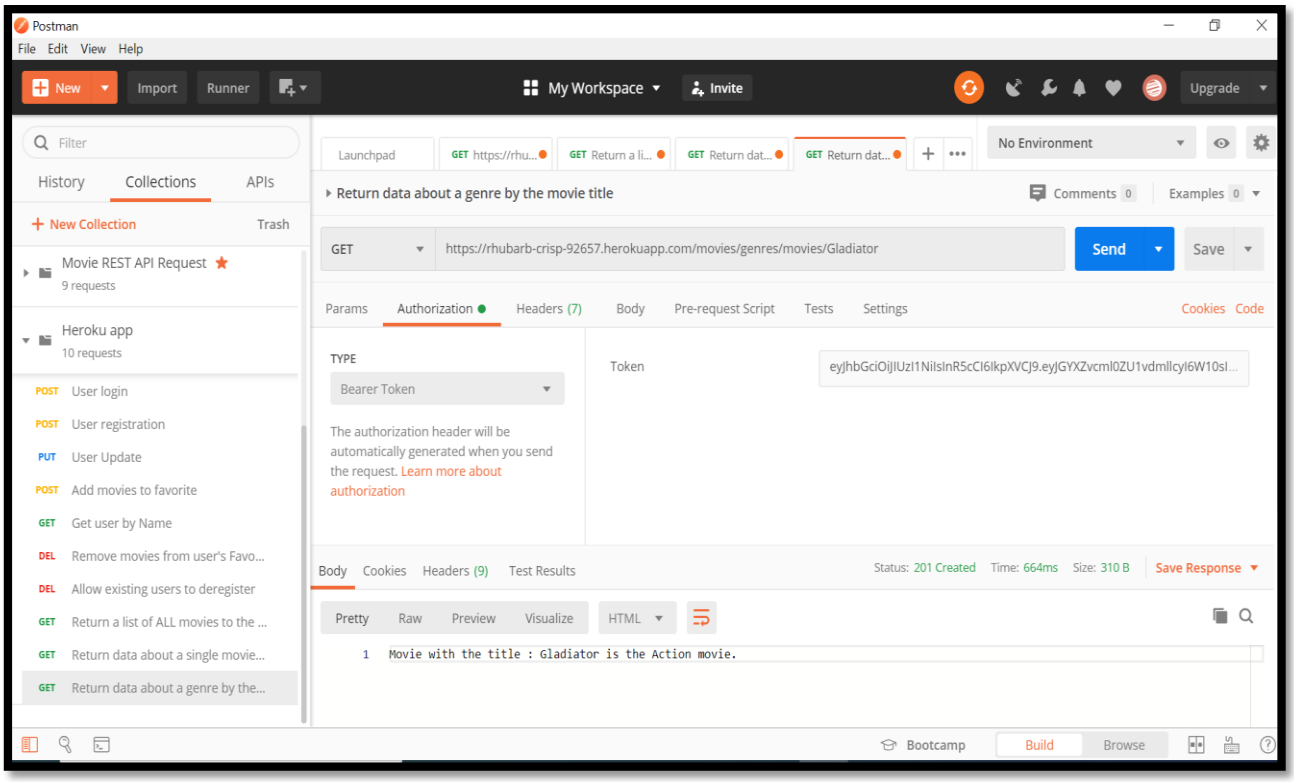
One: Return a list of ALL movies to the user:



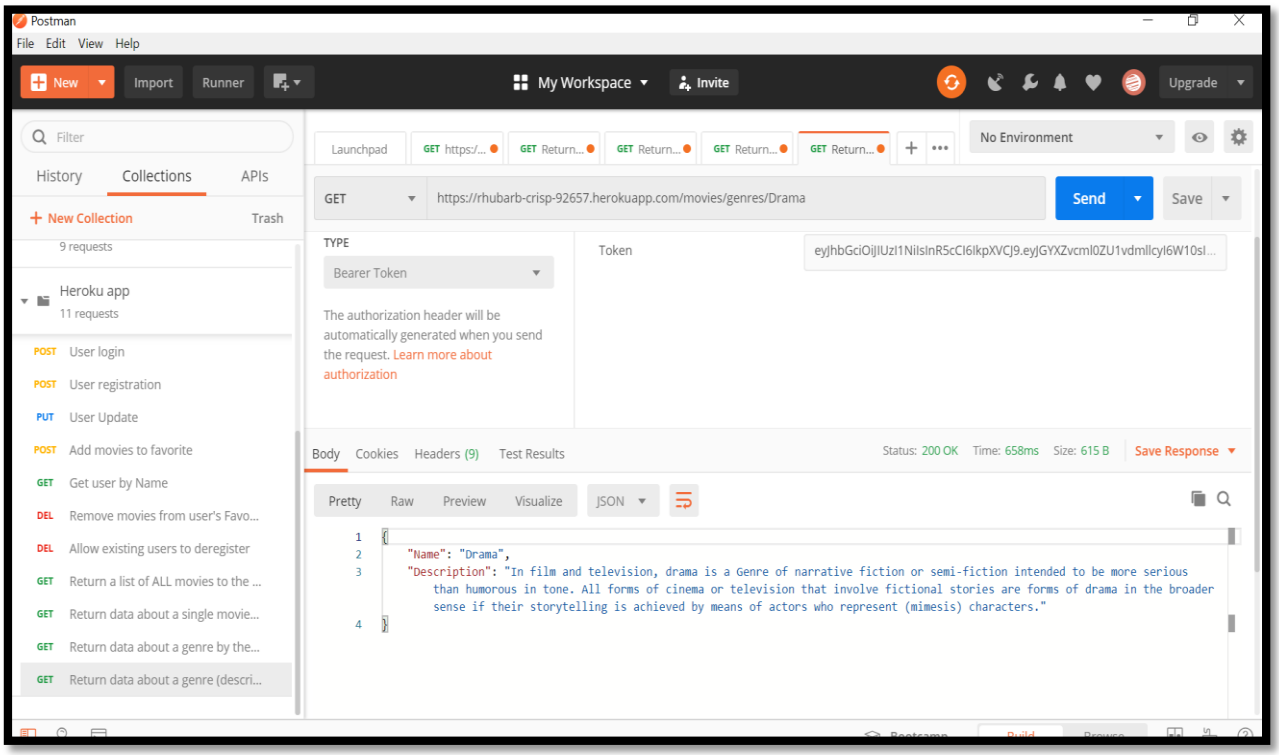
Two: Return data (description, genre, director, image URL, whether it's featured or not) about a single movie by title to the user:



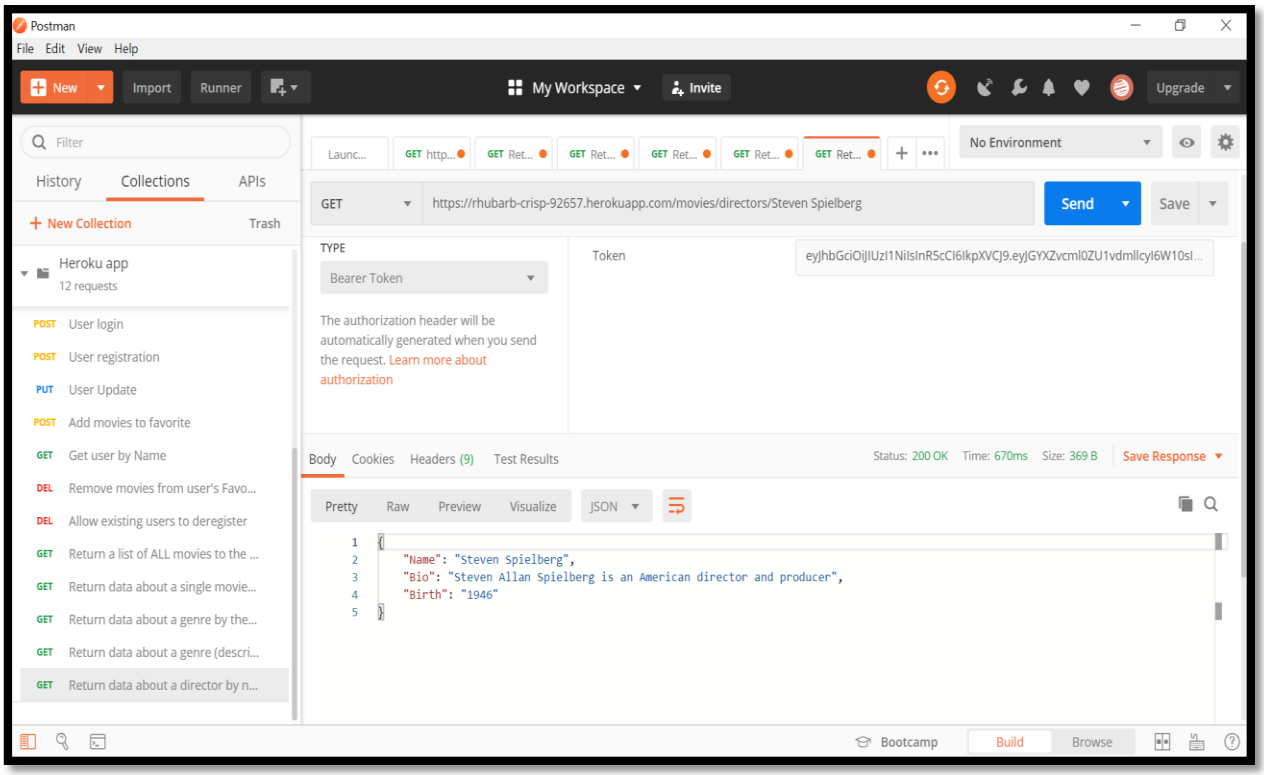
Three: Return data about a genre by the movie title (e.g., “Gladiator”):



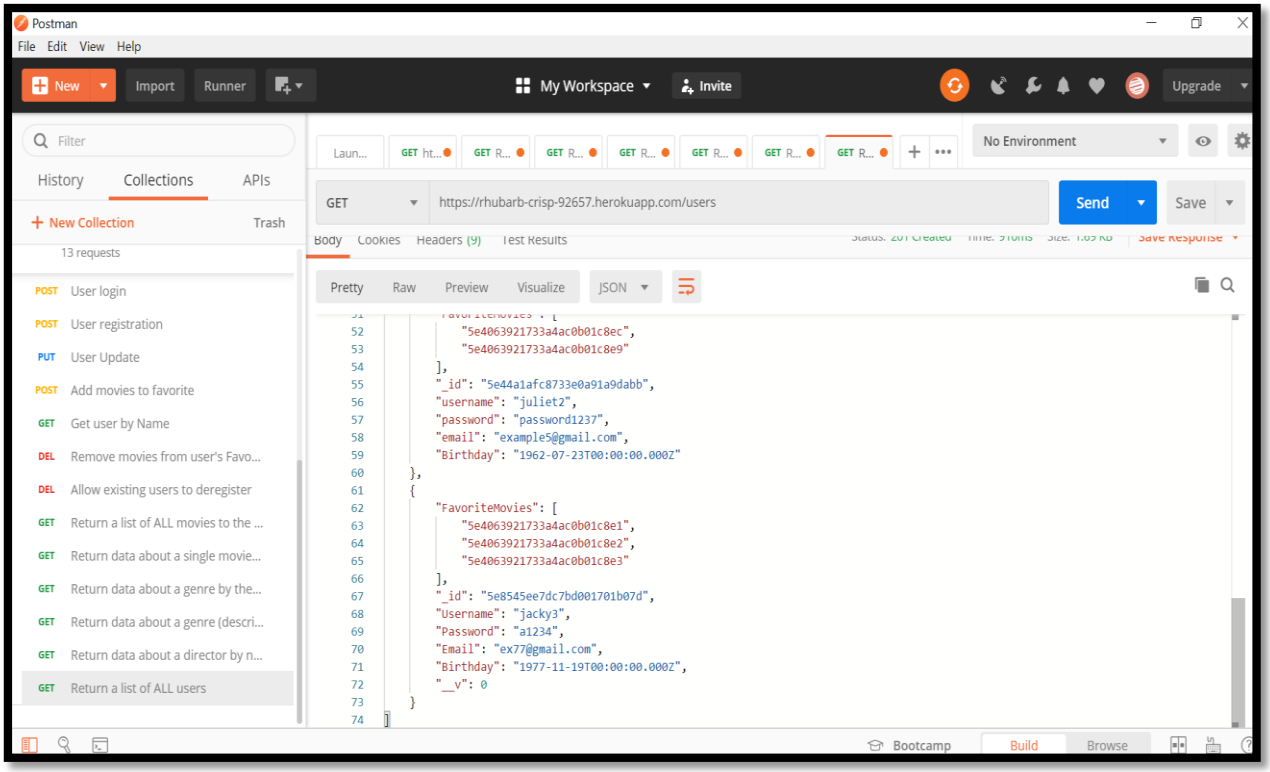
Four: Return data about a genre (description) by the genre Name (e.g., “Drama”):



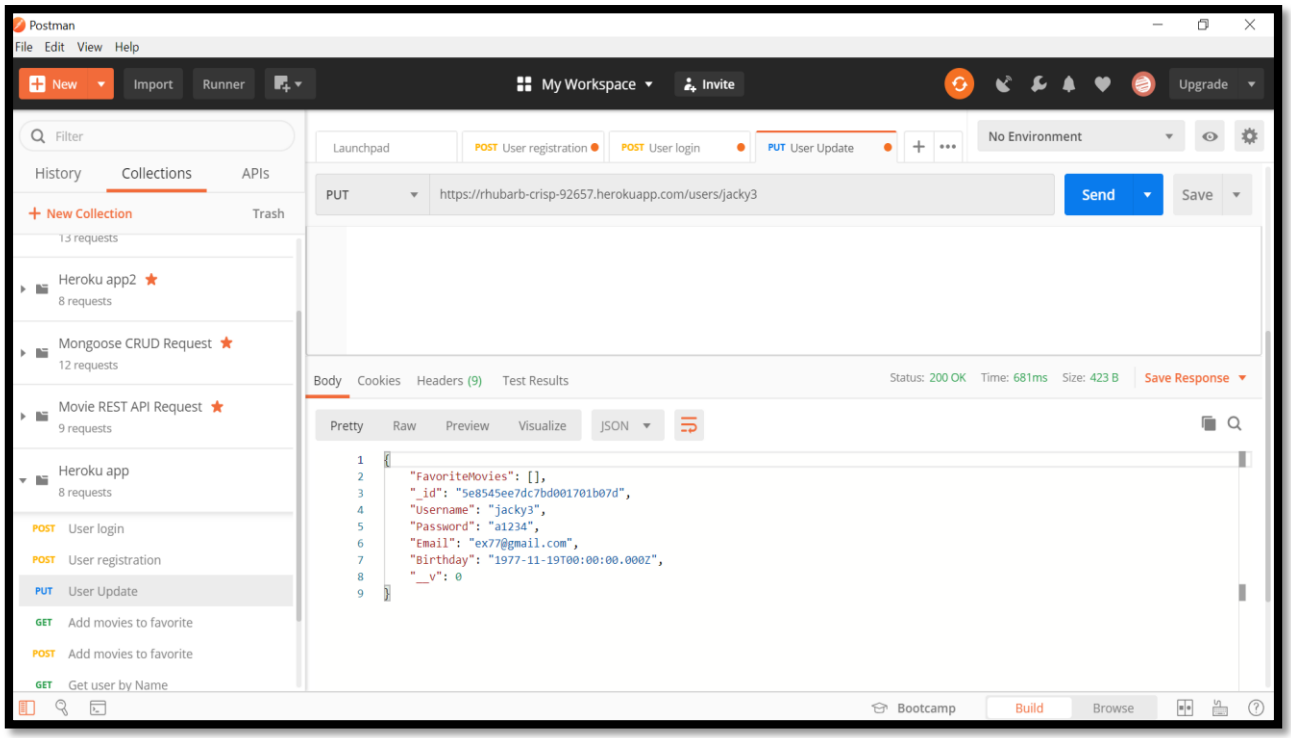
Five: Return data about a director (bio, birth year, death year) by name:



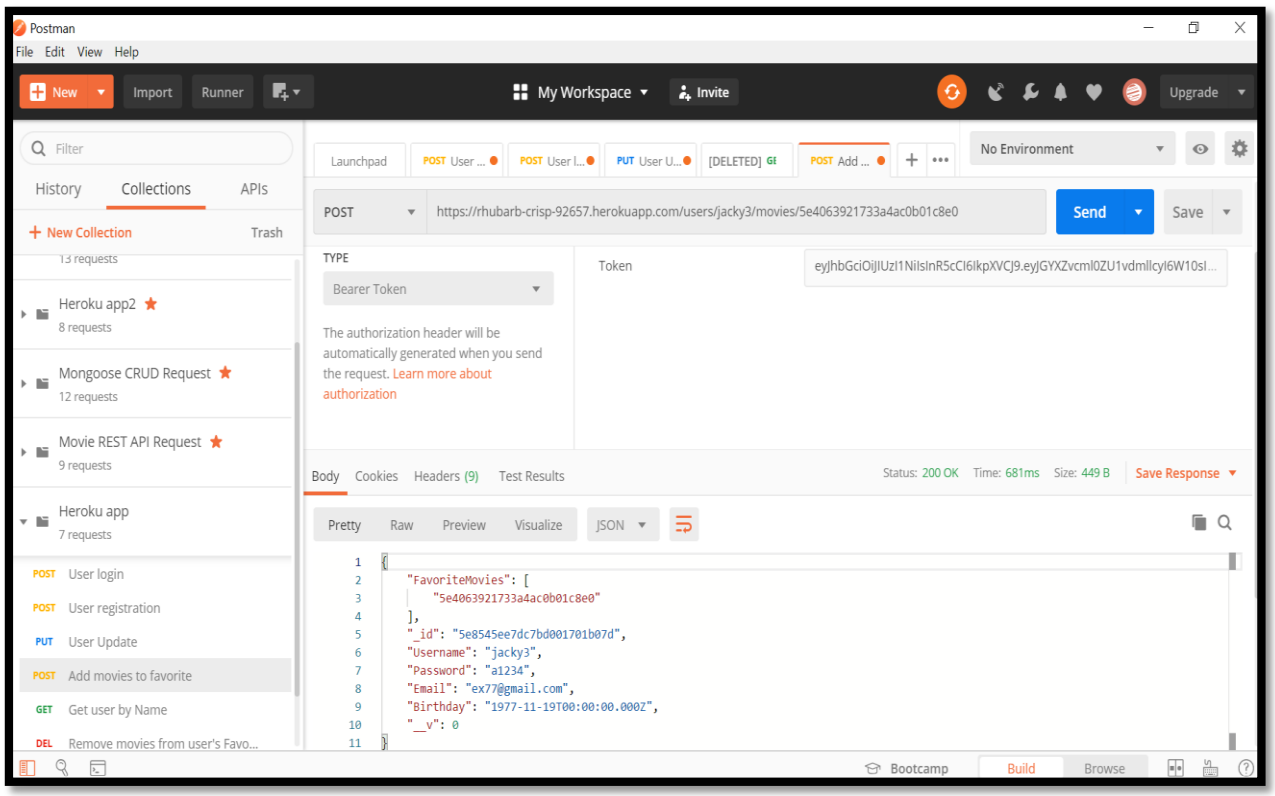
Six: Return a list of ALL users:



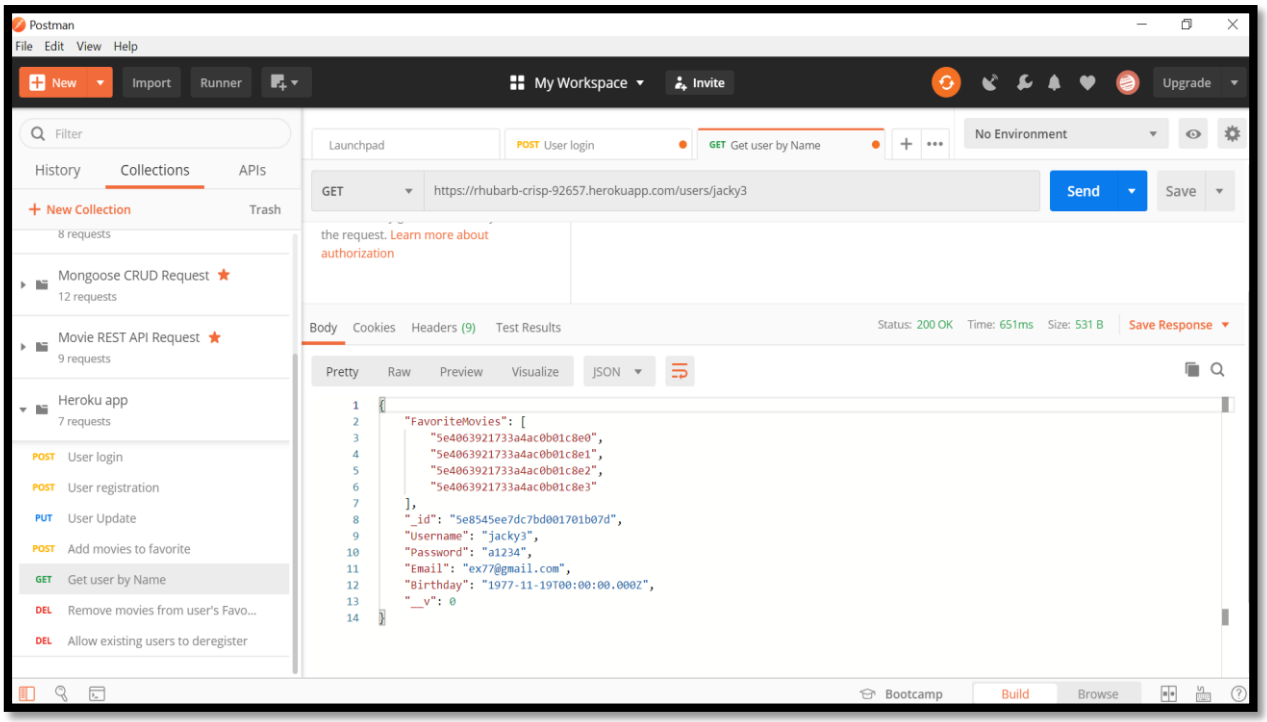
Seven: Allow users to update their user info (username, password, email, date of birth):



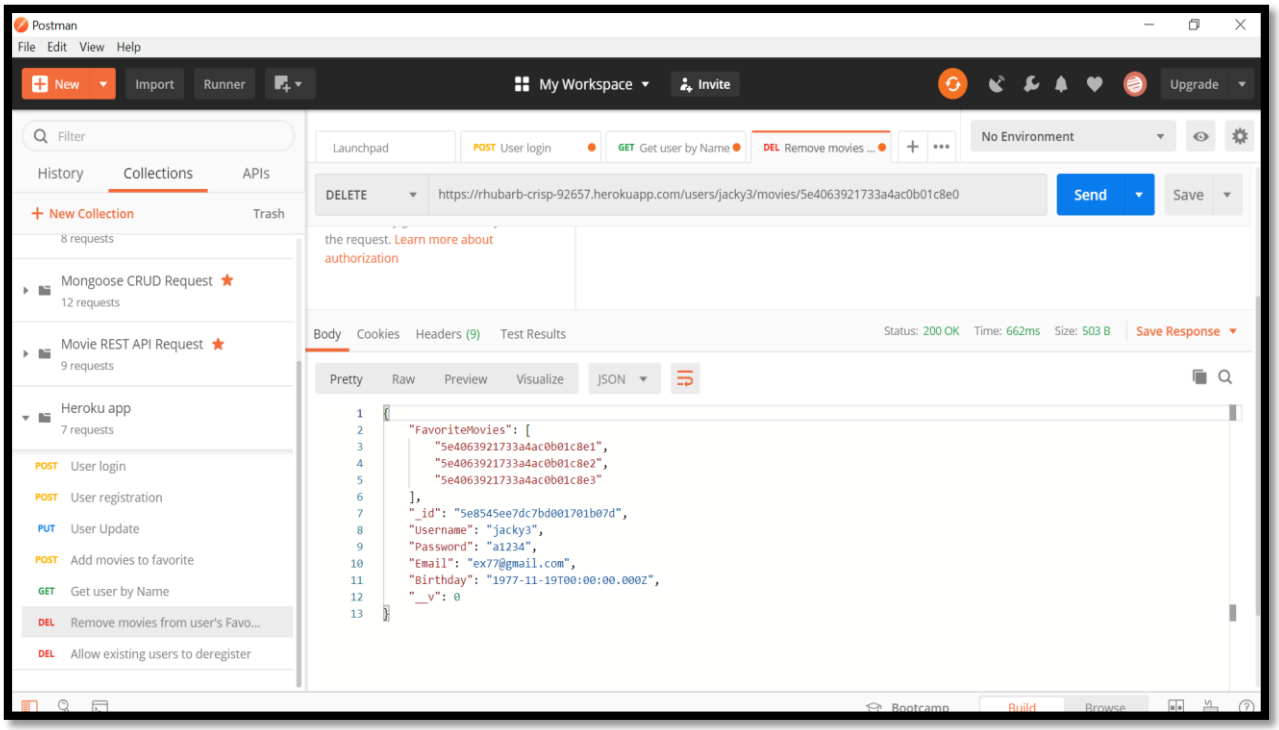
Eight: Allow users to add a movie to their list of favourites:



Nine: Get user by Name:



Ten: Allow users to remove a movie from their list of favourites:



Eleven: Allow existing users to deregister with authorization token: