Accessing MongoDB using Python

After completing this lab you will be able to:

- Access the MongoDB database from Python with the pymongo driver
- Perform basic operations such as selecting, inserting and listing using Python
- Create a Python program to run the MongoDB operations

Exercise 1 - Install the pymongo driver

```
theia@theiadocker-craigtrupp8:/home/project$ python3 -m pip install pymongo
Collecting pymongo
  Downloading
https://files.pythonhosted.org/packages/10/3b/46541b4ee3000019b8ef5b1847292
ddc77f492c162bc4d49c424db7fc97a/pymongo-4.1.1-cp36-cp36m-manylinux1_x86_64.
whl (464kB)
    100% |████████████████████████████████| 471kB 2.6MB/s
Installing collected packages: pymongo
Successfully installed pymongo-4.1.1
```

Exercise 2 - Start the server

```
theia@theiadocker-craigtrupp8:/home/project$ start_mongo
Starting your mongodb database....
This process can take up to a minute.

Mongodb started, waiting for all services to be ready....

Your mongodb server is now ready to use and available with username: root
password: NDAyNS1jcmFpZ3Ry

You can access your mongodb database via:
 • The browser at:
https://craigtrupp8-8081.theiadocker-2-labs-prod-theiak8s-4-tor01.proxy.cog
nitiveclass.ai
 • CommandLine: mongo -u root -p NDAyNS1jcmFpZ3Ry --authenticationDatabase
admin local
```

```
theia@theiadocker-craigtrupp8:/home/project$ mongo -u root -p
NDAyNS1jcmFpZ3Ry --authenticationDatabase admin local
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017/local
MongoDB server version: 3.6.3
Server has startup warnings:
2023-10-16T18:05:04.362+0000 I STORAGE  [initandlisten]
2023-10-16T18:05:04.362+0000 I STORAGE  [initandlisten] ** WARNING: Using
the XFS filesystem is strongly recommended with the WiredTiger storage
engine
2023-10-16T18:05:04.362+0000 I STORAGE  [initandlisten] **          See
http://dochub.mongodb.org/core/prodnotes-filesystem
2023-10-16T18:05:05.248+0000 I CONTROL  [initandlisten]
2023-10-16T18:05:05.248+0000 I CONTROL  [initandlisten] ** WARNING: You are
running on a NUMA machine.
2023-10-16T18:05:05.248+0000 I CONTROL  [initandlisten] **          We
suggest launching mongod like this to avoid performance problems:
2023-10-16T18:05:05.248+0000 I CONTROL  [initandlisten] **
numactl --interleave=all mongod [other options]
2023-10-16T18:05:05.248+0000 I CONTROL  [initandlisten]
```

Exercise 3 - Connect to mongodb server using Python

```python
from pymongo import MongoClient
user = 'root'
password = 'NDAyNS1jcmFpZ3Ry' #
host='localhost'
#create the connection url
connecturl =
"mongodb://{}:{}@{}:27017/?authSource=admin".format(user,password,host)

# connect to mongodb server
print("Connecting to mongodb server")
connection = MongoClient(connecturl)

# get database list
print("Getting list of databases")
dbs = connection.list_database_names()

# print the database names
```

```
for db in dbs:
    print(db)
print("Closing the connection to the mongodb server")

# close the server connecton
connection.close()
```

- Run the script

```
theia@theiadocker-craigtrupp8:/home/project$ python3 mongo_connect.py
Connecting to mongodb server
Getting list of databases
admin
local
Closing the connection to the mongodb server
```

Exercise 4 - Working with documents

- connect to the mongodb server.
- select a database named training.
- select a collection named python.
- insert a sample document.
- query all the documents in the training database and python collection.
- close the connection to the server.

```
from pymongo import MongoClient
user = 'root'
password = 'NDAyNS1jcmFpZ3Ry' # yourpw
host='localhost'
#create the connection url
connecturl =
"mongodb://{}:{}@{}:27017/?authSource=admin".format(user,password,host)

# connect to mongodb server
print("Connecting to mongodb server")
connection = MongoClient(connecturl)
```

```
# select the 'training' database

db = connection.training

# select/create the 'python' collection

collection = db.python

# create a sample document

doc = {"lab":"Accessing mongodb using python", "Subject":"No SQL
Databases"}

# insert a sample document

print("Inserting a document into collection.")
db.collection.insert_one(doc)

# query for all documents in 'training' database and 'python' collection

docs = db.collection.find()

print("Printing the documents in the collection.")

for document in docs:
    print(document)

# close the server connecton
print("Closing the connection.")
connection.close()
```

- Creation and running of file with connected instance to db

```
theia@theiadocker-craigtrupp8:/home/project$ touch mongo_query.py
theia@theiadocker-craigtrupp8:/home/project$ python3 mongo_query.py
Connecting to mongodb server
Inserting a document into collection.
Printing the documents in the collection.
{'_id': ObjectId('652d7cf5d9244eac2302c680'), 'lab': 'Accessing mongodb
using python', 'Subject': 'No SQL Databases'}
Closing the connection.
```

```
theia@theiadocker-craigtrupp8:/home/project$
```

Practice Exercise

Write a Python program that can:

- connect to the mongodb server.
- select/create a database named training.
- select a collection named mongodb_glossary.
- insert the following documents into the collection mongodb_glossary.
    - {“database”:”a database contains collections”}
    - {“collection”:”a collection stores the documents”}
    - {“document”:”a document contains the data in the form of key value pairs.”}

- query and print all the documents in the training database and mongodb_glossary collection.
- close the connection to the server.

```
from pymongo import MongoClient
user = 'root'
password = 'NDAyNS1jcmFpZ3Ry' # yourpw
host='localhost'
#create the connection url
connecturl =
"mongodb://{}:{}@{}:27017/?authSource=admin".format(user,password,host)


# connect to mongodb server
print("Connecting to mongodb server")
connection = MongoClient(connecturl)

# select/create the 'training' database
training_db = connection.training

# select/create a collection named mongodb_glossary
mng_db_glsry = training_db.mongodb_glossary

# insert a few documents into created collection
mng_db_insertMany = [{"database": "a database named collections"},
                     {"collection": "a collection stores the documents"},
                     {"document":"a document contains the data in the form
of key value pairs"}]

# insert the above documents
mng_db_glsry.insert_many(mng_db_insertMany)

# Query and print all documents in the training db and mongodb_glossary
collection
glossary_docs = mng_db_glsry.find()

for doc in glossary_docs:
    print(doc)

# close connection
print("Closing the connection")
connection.close()
```

- Ran in active mongodb collection

```
theia@theiadocker-craigtrupp8:/home/project$ python3 mongo_practice_ex.py
Connecting to mongodb server
{'_id': ObjectId('652d7f480b10266c5f1e7d90'), 'database': 'a database named
collections'}
{'_id': ObjectId('652d7f480b10266c5f1e7d91'), 'collection': 'a collection
stores the documents'}
{'_id': ObjectId('652d7f480b10266c5f1e7d92'), 'document': 'a document
contains the data in the form of key value pairs'}
Closing the connection
theia@theiadocker-craigtrupp8:/home/project$
```