(/)

Curriculum

Short Specializations

Average: **97.3%**

0x01. NoSQL

Back-end NoSQL MongoDB

By: Emmanuel Turlay, Staff Software Engineer at Cruise and Guillaume, CTO at Holberton school

Weight: 1

Project over - took place from Dec 18, 2023 6:00 AM to Dec 20, 2023 6:00 AM

☑ An auto review will be launched at the deadline

In a nutshell…

- **Auto QA review:** 79.95/135 mandatory & 0.0/32 optional
- **Altogether: 59.22%**
  - Mandatory: 59.22%
  - Optional: 0.0%
  - Calculation: 59.22% + (59.22% * 0.0%) == **59.22%**

Resources

Read or watch:

- NoSQL Databases Explained (/rltoken/wweK7dOY4pf8haCqv9Iv6Q)
- What is NoSQL ? (/rltoken/QqqNmgzgwopHBv305ki6bg)
- MongoDB with Python Crash Course - Tutorial for Beginners (/rltoken/RyyP9OH1EMBWWYpTs4TqoA)
- MongoDB Tutorial 2 : Insert, Update, Remove, Query (/rltoken/9__3tR-NimgXlmjPQwTF-Q)
- Aggregation (/rltoken/ziEDeniRobC6owPE1_avAQ)
- Introduction to MongoDB and Python (/rltoken/axwwF4CjO7FnK8Ecochqnw)
- mongo Shell Methods (/rltoken/lUqnLwOHbbp9FK39ijNmDQ)
- Mongosh (/rltoken/ipHIVVmAsezINqpk7W0eow)

Learning Objectives

(/)

At the end of this project, you are expected to be able to explain to anyone (/rltoken/9u20uNESC1dnTNowO5waNQ), **without the help of Google**:

General

- What NoSQL means
- What is difference between SQL and NoSQL
- What is ACID
- What is a document storage
- What are NoSQL types
- What are benefits of a NoSQL database
- How to query information from a NoSQL database
- How to insert/update/delete information from a NoSQL database
- How to use MongoDB

Requirements

MongoDB Command File

- All your files will be interpreted/compiled on Ubuntu 18.04 LTS using `MongoDB` (version 4.2)
- All your files should end with a new line
- The first line of all your files should be a comment: `// my comment`
- A `README.md` file, at the root of the folder of the project, is mandatory
- The length of your files will be tested using `wc`

Python Scripts

- All your files will be interpreted/compiled on Ubuntu 18.04 LTS using `python3` (version 3.7) and `PyMongo` (version 3.10)
- All your files should end with a new line
- The first line of all your files should be exactly `#!/usr/bin/env python3`
- A `README.md` file, at the root of the folder of the project, is mandatory
- Your code should use the `pycodestyle` style (version 2.5.*)
- The length of your files will be tested using `wc`
- All your modules should have a documentation (`python3 -c 'print(__import__("my_module").__doc__)'`)
- All your functions should have a documentation (`python3 -c 'print(__import__("my_module").my_function.__doc__)'`
- Your code should not be executed when imported (by using `if __name__ == "__main__"` :)

More Info

Install MongoDB 4.2 in Ubuntu 18.04

Official installation guide (/rltoken/8p4x14Ddn1UxKXZ5nPt3zA)

```
$ wget -qO - https://www.mongodb.org/static/pgp/server-4.2.asc | apt-key add -
$ echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.2 multiverse" > /etc/apt/sources.list.d/mongodb-org-4.2.list
$ sudo apt-get update
$ sudo apt-get install -y mongodb-org
...
$  sudo service mongod status
mongod start/running, process 3627
$ mongo --version
MongoDB shell version v4.2.8
git version: 43d25964249164d76d5e04dd6cf38f6111e21f5f
OpenSSL version: OpenSSL 1.1.1  11 Sep 2018
allocator: tcmalloc
modules: none
build environment:
    distmod: ubuntu1804
    distarch: x86_64
    target_arch: x86_64
$
$ pip3 install pymongo
$ python3
>>> import pymongo
>>> pymongo.__version__
'3.10.1'
```

Potential issue if documents creation doesn’t work or this error: `Data directory /data/db not found., terminating` (source (/rltoken/as8vd5VBnj4VDz5EINszMg) and source (/rltoken/9Df5v1NcWFFCn_sRNgsJUg))

```
$ sudo mkdir -p /data/db
```

Or if `/etc/init.d/mongod` is missing, please find here an example of the file:

Click to expand/hide file contents

Use “container-on-demand” to run MongoDB

- Ask for container `Ubuntu 18.04 - MongoDB`
- Connect via SSH
- Or via the WebTerminal
- In the container, you should start MongoDB before playing with it:

```
$ service mongod start
 * Starting database mongod                                               [ OK ]
$
$ cat 0-list_databases | mongo
MongoDB shell version v4.2.8
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mon
godb
Implicit session: session { "id" : UUID("70f14b38-6d0b-48e1-a9a4-0534bcf15301") }
MongoDB server version: 4.2.8
admin   0.000GB
config  0.000GB
local   0.000GB
bye
$
```

Tasks

0. List all databases

mandatory

Score: 65.0% (*Checks completed: 100.0%*)

Write a script that lists all databases in MongoDB.

```
guillaume@ubuntu:~/0x01$ cat 0-list_databases | mongo
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.3
admin        0.000GB
config       0.000GB
local        0.000GB
logs         0.005GB
bye
guillaume@ubuntu:~/0x01$
```

Repo:

- GitHub repository: `alx-backend-storage`
- Directory: `0x01-NoSQL`
- File: `0-list_databases`

☑ Done! Help Check your code Get a sandbox QA Review

1. Create a database

mandatory

Score: 65.0% (*Checks completed: 100.0%*)

Write a script that creates or uses the database `my_db` :

```
guillaume@ubuntu:~/0x01$ cat 0-list_databases | mongo
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.3
admin        0.000GB
config       0.000GB
local        0.000GB
logs         0.005GB
bye
guillaume@ubuntu:~/0x01$
guillaume@ubuntu:~/0x01$ cat 1-use_or_create_database | mongo
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.3
switched to db my_db
bye
guillaume@ubuntu:~/0x01$
```

Repo:

- GitHub repository: `alx-backend-storage`
- Directory: `0x01-NoSQL`
- File: `1-use_or_create_database`

☑ Done! | Help | Check your code | Get a sandbox | QA Review

2. Insert document

mandatory

Score: 65.0% (*Checks completed: 100.0%*)

Write a script that inserts a document in the collection `school` :

- The document must have one attribute `name` with value “Holberton school”
- The database name will be passed as option of `mongo` command

```
guillaume@ubuntu:~/0x01$ cat 2-insert | mongo my_db
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017/my_db
MongoDB server version: 3.6.3
WriteResult({ "nInserted" : 1 })
bye
guillaume@ubuntu:~/0x01$
```

Repo:

- GitHub repository: `alx-backend-storage`
- Directory: `0x01-NoSQL`
- File: `2-insert`

☑ Done! | Help | Check your code | >_ Get a sandbox | QA Review

3. All documents

mandatory

Score: 65.0% (*Checks completed: 100.0%*)

Write a script that lists all documents in the collection `school`:

- The database name will be passed as option of `mongo` command

```
guillaume@ubuntu:~/0x01$ cat 3-all | mongo my_db
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017/my_db
MongoDB server version: 3.6.3
{ "_id" : ObjectId("5a8fad532b69437b63252406"), "name" : "Holberton school" }
bye
guillaume@ubuntu:~/0x01$
```

Repo:

- GitHub repository: `alx-backend-storage`
- Directory: `0x01-NoSQL`
- File: `3-all`

☑ Done! | Help | Check your code | >_ Get a sandbox | QA Review

(/)

4. All matches

mandatory

Score: 65.0% (*Checks completed: 100.0%*)

Write a script that lists all documents with `name="Holberton school"` in the collection `school`:

- The database name will be passed as option of `mongo` command

```
guillaume@ubuntu:~/0x01$ cat 4-match | mongo my_db
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017/my_db
MongoDB server version: 3.6.3
{ "_id" : ObjectId("5a8fad532b69437b63252406"), "name" : "Holberton school" }
bye
guillaume@ubuntu:~/0x01$
```

Repo:

- GitHub repository: `alx-backend-storage`
- Directory: `0x01-NoSQL`
- File: `4-match`

☑ Done! Help Check your code >_ Get a sandbox QA Review

5. Count

mandatory

Score: 65.0% (*Checks completed: 100.0%*)

Write a script that displays the number of documents in the collection `school`:

- The database name will be passed as option of `mongo` command

```
guillaume@ubuntu:~/0x01$ cat 5-count | mongo my_db
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017/my_db
MongoDB server version: 3.6.3
1
bye
guillaume@ubuntu:~/0x01$
```

Repo:

- GitHub repository: `alx-backend-storage`
- Directory: `0x01-NoSQL`

- File: `5-count`

(/)

☑ Done! Help Check your code >_ Get a sandbox QA Review

6. Update

mandatory

Score: 65.0% (*Checks completed: 100.0%*)

Write a script that adds a new attribute to a document in the collection `school`:

- The script should update only document with `name="Holberton school"` (all of them)
- The update should add the attribute `address` with the value “972 Mission street”
- The database name will be passed as option of `mongo` command

```
guillaume@ubuntu:~/0x01$ cat 6-update | mongo my_db
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017/my_db
MongoDB server version: 3.6.3
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
bye
guillaume@ubuntu:~/0x01$
guillaume@ubuntu:~/0x01$ cat 4-match | mongo my_db
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017/my_db
MongoDB server version: 3.6.3
{ "_id" : ObjectId("5a8fad532b69437b63252406"), "name" : "Holberton school", "addres
s" : "972 Mission street" }
bye
guillaume@ubuntu:~/0x01$
```

Repo:

- GitHub repository: `alx-backend-storage`
- Directory: `0x01-NoSQL`
- File: `6-update`

☑ Done! Help Check your code >_ Get a sandbox QA Review

7. Delete by match

mandatory

Score: 65.0% (*Checks completed: 100.0%*)

Write a script that deletes all documents with `name="Holberton school"` in the collection `school`:
(/)

- The database name will be passed as option of `mongo` command

```
guillaume@ubuntu:~/0x01$ cat 7-delete | mongo my_db
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017/my_db
MongoDB server version: 3.6.3
{ "acknowledged" : true, "deletedCount" : 1 }
bye
guillaume@ubuntu:~/0x01$
guillaume@ubuntu:~/0x01$ cat 4-match | mongo my_db
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017/my_db
MongoDB server version: 3.6.3
bye
guillaume@ubuntu:~/0x01$
```

Repo:

- GitHub repository: `alx-backend-storage`
- Directory: `0x01-NoSQL`
- File: `7-delete`

☑ Done! | Help | Check your code | >_ Get a sandbox | QA Review

8. List all documents in Python

mandatory

Score: 65.0% (*Checks completed: 100.0%*)

Write a Python function that lists all documents in a collection:

- Prototype: `def list_all(mongo_collection):`
- Return an empty list if no document in the collection
- `mongo_collection` will be the `pymongo` collection object

```
guillaume@ubuntu:~/0x01$ cat 8-main.py
#!/usr/bin/env python3
""" 8-main """
from pymongo import MongoClient
list_all = __import__('8-all').list_all

if __name__ == "__main__":
    client = MongoClient('mongodb://127.0.0.1:27017')
    school_collection = client.my_db.school
    schools = list_all(school_collection)
    for school in schools:
        print("[{}] {}".format(school.get('_id'), school.get('name')))

guillaume@ubuntu:~/0x01$
guillaume@ubuntu:~/0x01$ ./8-main.py
[5a8f60cfd4321e1403ba7ab9] Holberton school
[5a8f60cfd4321e1403ba7aba] UCSD
guillaume@ubuntu:~/0x01$
```

Repo:

- GitHub repository: `alx-backend-storage`
- Directory: `0x01-NoSQL`
- File: `8-all.py`

☑ Done! Help Check your code Get a sandbox QA Review

9. Insert a document in Python

mandatory

Score: 65.0% (*Checks completed: 100.0%*)

Write a Python function that inserts a new document in a collection based on `kwargs`:

- Prototype: `def insert_school(mongo_collection, **kwargs):`
- `mongo_collection` will be the `pymongo` collection object
- Returns the new `_id`

```
guillaume@ubuntu:~/0x01$ cat 9-main.py
#!/usr/bin/env python3
""" 9-main """
from pymongo import MongoClient
list_all = __import__('8-all').list_all
insert_school = __import__('9-insert_school').insert_school

if __name__ == "__main__":
    client = MongoClient('mongodb://127.0.0.1:27017')
    school_collection = client.my_db.school
    new_school_id = insert_school(school_collection, name="UCSF", address="505 Parna
ssus Ave")
    print("New school created: {}".format(new_school_id))

    schools = list_all(school_collection)
    for school in schools:
        print("[{}] {} {}".format(school.get('_id'), school.get('name'), school.get
('address', "")))

guillaume@ubuntu:~/0x01$
guillaume@ubuntu:~/0x01$ ./9-main.py
New school created: 5a8f60cfd4321e1403ba7abb
[5a8f60cfd4321e1403ba7ab9] Holberton school
[5a8f60cfd4321e1403ba7aba] UCSD
[5a8f60cfd4321e1403ba7abb] UCSF 505 Parnassus Ave
guillaume@ubuntu:~/0x01$
```

Repo:

- GitHub repository: `alx-backend-storage`
- Directory: `0x01-NoSQL`
- File: `9-insert_school.py`

☑ Done! Help Check your code >_ Get a sandbox QA Review

10. Change school topics

mandatory

Score: 65.0% (*Checks completed: 100.0%*)

Write a Python function that changes all topics of a school document based on the name:

- Prototype: `def update_topics(mongo_collection, name, topics):`
- `mongo_collection` will be the `pymongo` collection object
- `name` (string) will be the school name to update
- `topics` (list of strings) will be the list of topics approached in the school

```
guillaume@ubuntu:~/0x01$ cat 10-main.py
#!/usr/bin/env python3
""" 10-main """
from pymongo import MongoClient
list_all = __import__('8-all').list_all
update_topics = __import__('10-update_topics').update_topics

if __name__ == "__main__":
    client = MongoClient('mongodb://127.0.0.1:27017')
    school_collection = client.my_db.school
    update_topics(school_collection, "Holberton school", ["Sys admin", "AI", "Algorithm"])

    schools = list_all(school_collection)
    for school in schools:
        print("[{}] {} {}".format(school.get('_id'), school.get('name'), school.get('topics', "")))

    update_topics(school_collection, "Holberton school", ["iOS"])

    schools = list_all(school_collection)
    for school in schools:
        print("[{}] {} {}".format(school.get('_id'), school.get('name'), school.get('topics', "")))

guillaume@ubuntu:~/0x01$
guillaume@ubuntu:~/0x01$ ./10-main.py
[5a8f60cfd4321e1403ba7abb] UCSF
[5a8f60cfd4321e1403ba7aba] UCSD
[5a8f60cfd4321e1403ba7ab9] Holberton school ['Sys admin', 'AI', 'Algorithm']
[5a8f60cfd4321e1403ba7abb] UCSF
[5a8f60cfd4321e1403ba7aba] UCSD
[5a8f60cfd4321e1403ba7ab9] Holberton school ['iOS']
guillaume@ubuntu:~/0x01$
```

Repo:

- GitHub repository: `alx-backend-storage`
- Directory: `0x01-NoSQL`
- File: `10-update_topics.py`

☑ Done! Help Check your code Get a sandbox QA Review

11. Where can I learn Python?

mandatory

Score: 65.0% (*Checks completed: 100.0%*)

Write a Python function that returns the list of school having a specific topic:

- Prototype: `def schools_by_topic(mongo_collection, topic):`
- `mongo_collection` will be the `pymongo` collection object
- `topic` (string) will be topic searched

```
guillaume@ubuntu:~/0x01$ cat 11-main.py
#!/usr/bin/env python3
""" 11-main """
from pymongo import MongoClient
list_all = __import__('8-all').list_all
insert_school = __import__('9-insert_school').insert_school
schools_by_topic = __import__('11-schools_by_topic').schools_by_topic

if __name__ == "__main__":
    client = MongoClient('mongodb://127.0.0.1:27017')
    school_collection = client.my_db.school

    j_schools = [
        { 'name': "Holberton school", 'topics': ["Algo", "C", "Python", "React"]},
        { 'name': "UCSF", 'topics': ["Algo", "MongoDB"]},
        { 'name': "UCLA", 'topics': ["C", "Python"]},
        { 'name': "UCSD", 'topics': ["Cassandra"]},
        { 'name': "Stanford", 'topics': ["C", "React", "Javascript"]}
    ]
    for j_school in j_schools:
        insert_school(school_collection, **j_school)

    schools = schools_by_topic(school_collection, "Python")
    for school in schools:
        print("[{}] {} {}".format(school.get('_id'), school.get('name'), school.get
('topics', "")))

guillaume@ubuntu:~/0x01$
guillaume@ubuntu:~/0x01$ ./11-main.py
[5a90731fd4321e1e5a3f53e3] Holberton school ['Algo', 'C', 'Python', 'React']
[5a90731fd4321e1e5a3f53e5] UCLA ['C', 'Python']
guillaume@ubuntu:~/0x01$
```

Repo:

- GitHub repository: `alx-backend-storage`
- Directory: `0x01-NoSQL`
- File: `11-schools_by_topic.py`

☑ Done! | Help | Check your code | >_ Get a sandbox | QA Review

12. Log stats

mandatory

Write a Python script that provides some stats about Nginx logs stored in MongoDB:

Score: 0.0% (*Checks completed: 0.0%*)

(/)

- Database: `logs`
- Collection: `nginx`
- Display (same as the example):
  - first line: `x logs` where `x` is the number of documents in this collection
  - second line: `Methods:`
  - 5 lines with the number of documents with the `method = ["GET", "POST", "PUT", "PATCH", "DELETE"]` in this order (see example below - warning: it’s a tabulation before each line)
  - one line with the number of documents with:
    - `method=GET`
    - `path=/status`

You can use this dump as data sample: dump.zip (/rltoken/0szbpslKvH3RqKb_2HUeoQ)

The output of your script **must be exactly the same as the example**

```
guillaume@ubuntu:~/0x01$ curl -o dump.zip -s "https://s3.amazonaws.com/intranet-projects-files/holbertonschool-webstack/411/dump.zip"
guillaume@ubuntu:~/0x01$ 
guillaume@ubuntu:~/0x01$ unzip dump.zip
Archive:  dump.zip
   creating: dump/
   creating: dump/logs/
  inflating: dump/logs/nginx.metadata.json
  inflating: dump/logs/nginx.bson
guillaume@ubuntu:~/0x01$ 
guillaume@ubuntu:~/0x01$ mongorestore dump
2018-02-23T20:12:37.807+0000    preparing collections to restore from
2018-02-23T20:12:37.816+0000    reading metadata for logs.nginx from dump/logs/nginx.metadata.json
2018-02-23T20:12:37.825+0000    restoring logs.nginx from dump/logs/nginx.bson
2018-02-23T20:12:40.804+0000    [##......................]  logs.nginx  1.21MB/13.4MB  (9.0%)
2018-02-23T20:12:43.803+0000    [#####...................]  logs.nginx  2.88MB/13.4MB  (21.4%)
2018-02-23T20:12:46.803+0000    [#######.................]  logs.nginx  4.22MB/13.4MB  (31.4%)
2018-02-23T20:12:49.803+0000    [##########..............]  logs.nginx  5.73MB/13.4MB  (42.7%)
2018-02-23T20:12:52.803+0000    [############............]  logs.nginx  7.23MB/13.4MB  (53.8%)
2018-02-23T20:12:55.803+0000    [###############.........]  logs.nginx  8.53MB/13.4MB  (63.5%)
2018-02-23T20:12:58.803+0000    [#################.......]  logs.nginx  10.1MB/13.4MB  (74.9%)
2018-02-23T20:13:01.803+0000    [####################....]  logs.nginx  11.3MB/13.4MB  (83.9%)
2018-02-23T20:13:04.803+0000    [######################..]  logs.nginx  12.8MB/13.4MB  (94.9%)
2018-02-23T20:13:06.228+0000    [########################]  logs.nginx  13.4MB/13.4MB  (100.0%)
2018-02-23T20:13:06.230+0000    no indexes to restore
2018-02-23T20:13:06.231+0000    finished restoring logs.nginx (94778 documents)
2018-02-23T20:13:06.232+0000    done
guillaume@ubuntu:~/0x01$ 
guillaume@ubuntu:~/0x01$ ./12-log_stats.py 
94778 logs
Methods:
    method GET: 93842
    method POST: 229
    method PUT: 0
    method PATCH: 0
    method DELETE: 0
47415 status check
guillaume@ubuntu:~/0x01$ 
```

Repo:

- GitHub repository: `alx-backend-storage`
- Directory: `0x01-NoSQL`
- File: `12-log_stats.py`

☐ Done? Help Check your code Ask for a new correction >_ Get a sandbox QA Review

Done with the mandatory tasks? Unlock 3 advanced tasks now!