 Home(/)

0x01. NoSQL

 Check-end

 Plan

 MongoDB



Weight: 1

Projects(/projects/current)



Project over - took place from Jul 15, 2024 6:00 AM to Jul 17, 2024 6:00 AM



An auto QA review will automatically make corrections(/to_review)



Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

In a nutshell...

- **Auto QA review:** 123.0/135 mandatory & 21.0/32 optional



- **Altogether: 150.91%**

Curriculums(/dashboards/my_curriculums)

- Mandatory: 91.11%
- Optional: 65.63%



- Calculation: $91.11\% + (91.11\% * 65.63\%) == 150.91\%$

Concepts(/concepts)

Conference rooms(/dashboards/video_rooms)



Resources

Servers(/servers)

Read or watch:



- NoSQL Databases Explained (/rltoken/wweK7dOY4pf8haCqv9lv6Q)

- What is NoSQL ? (/rltoken/QqqNmgzgwopHBv305ki6bg)

- MongoDB with Python Crash Course - Tutorial for Beginners



- Tools for NoSQL Databases (/rltoken/8s4TqoA)

- 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/IUqnLwOHbbp9FK39ijNmDQ)

- Mongosh (/rltoken/ipHIVvmAseziNqpk7W0eow)



Learning Objectives

Peers(/users/peers)

At the end of this project, you are expected to be able to explain to anyone



Discord (/https://discord.com/app) without the help of Google:

General

- What NoSQL means
- What is difference between SQL and NoSQL
- What is ACID

My Profile(/users/my_profile)





- 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



Home(/)

Requirements

My Learning(/learning/me)

MongoDB Command File

Projects(/projects/current)

- 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`

Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

Python Scripts



- All your files will be interpreted/compiled on Ubuntu 18.04 LTS using python3 (version 3.7) and PyMongo (version 3.10)

Curriculums(/dashboards/my_curriculums)



- 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.*)

Concepts(/concepts)



- The reference your files shall be tested using `docs`
- 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__":`)

Sandboxes(/user_containers/current)

More Info

Tools(/dashboards/my_tools)



Install MongoDB 4.2 in Ubuntu 18.04

Video on demand(/dashboards/videos)

Official installation guide (/rltoken/8p4x14Ddn1UxKXZ5nPt3zA)








Peers(/users/peers)






Discord(<https://discord.com/app>)



My Profile(/users/my_profile)

 \$ 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/mongo-
db-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
 ... Home(/)
\$ sudo service mongod status
mongod start/running, process 3627
 My Planning(/planning/me)
MongoDB shell version v4.2.8
git version: 43d25964249164d76d5e04dd6cf38f6111e21f5f
 OpenSS (projects/projects/current) 1.1 11 Sep 2018
allocator: tcmalloc
modules: none
✓ build QA Reviews (can make(/corrections/to_review)
distmod: ubuntu1804
distarch: x86_64
? t-eva arch: x86_64
\$
\$ pip3 install pymongo
\$ python3
 >>> import pymongo
>>> pymongo.__version__
'3.10.1'

 Concepts(/concepts)
Potential issue if documents creation doesn't work or this error: Data directory /data/db not
found., terminating (source (/rltoken/as8vd5VBnj4VDz5EINszMg) and source
 (/rltoken/9Df5y1NcWEECn-sRNgsJLq))
Conference rooms(/dashboards/video_rooms)



 \$ sudo mkdir -p /data/db


 Servers(/servers)

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

 Click to expand/hide file contents
Sandboxes(/user_containers/current)

Use "container-on-demand" to run MongoDB

-  Tools(/dashboards/my_tools)
- Ask for container Ubuntu 18.04 - MongoDB
 - Connect via SSH
 -  • Or via the Web Terminal (dashboards/videos)
 - In the container, you should start MongoDB before playing with it:

 Peers(/users/peers)

 Discord(<https://discord.com/app>)



My Profile(/users/my_profile)

```

$ service mongod start
* Starting database mongod
$ (/)
$ cat 0-list_databases | mongo
MongoDB shell version v4.2.8
connecting to: mongod://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=
mongod
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
$
QA Reviews I can make(/corrections/to_review)

```

? Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

Tasks

 List all databases(/dashboards/my_curriculums)

mandatory

 Score: 100.0% (Checks completed: 100.0%)
Concepts(/concepts)

Write a script that lists all databases in MongoDB.

```

Conference rooms(/dashboards/video_rooms)
guillaume@ubuntu:~/0x01$ cat 0-list_databases | mongo
MongoDB shell version v3.6.3
connecting to: mongod://127.0.0.1:27017
MongoDB server version: 3.6.3
admin    0.000GB
config   0.000GB
local    0.000GB
logs     0.005GB
bye
Tools(/dashboards/my_tools)
guillaume@ubuntu:~/0x01$

```

 Video on demand(/dashboards/videos)

Repo:

- GitHub repository: alx-backend-storage
- Directory: 0x01-NoSQL Peers(/users/peers)
- File: 0-list_databases

 Discord(<https://discord.com/app>)

Check submission

> Get a sandbox

View results



My Profile(/users/my_profile)

1. Create a database

mandatory

(/)

Score: 100.0% (Checks completed: 100.0%)



Home(/)

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 QA Reviews I can make(/corrections/to_review) 0.003GB
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
Curriculums(/dashboards/my_curriculums)
bye
guillaume@ubuntu:~/0x01$
```



Concepts(/concepts)

Repo:



Conference rooms(/dashboards/video_rooms)

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



- Servers(/servers)



Check summaries(/user_containers/current) results



2. Insert document

Tools(/dashboards/my_tools)

mandatory

Score: 100.0% (Checks completed: 100.0%)



Video on demand(/dashboards/videos)

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$
```

My Profile(/users/my_profile)



Repo:



- GitHub repository: alx-backend-storage
- Directory: 0x01-NoSQL
- File: 2-insert Home(/)



Check submission



Get a sandbox

View results

My Planning(/planning/me)

3. All documents

mandatory



Projects(/projects/current)

Score: 100.0% (Checks completed: 100.0%)



QA Reviews I can make(/corrections/to_review)

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



- The database name will be passed as option of mongo command
Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

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



Curricula(/dashboards/my_curriculums)



Concepts(/concepts)



Conference rooms(/dashboards/video_rooms)

Repo:



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



Sandboxes(/user_containers/current)

Check submission



Get a sandbox

View results



Tools(/dashboards/my_tools)



Video on demand(/dashboards/videos)

4. All matches

mandatory

Score: 100.0% (Checks completed: 100.0%)



Peers(/users/peers)

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
Discord(<https://discord.com/app>)





My Profile(/users/my_profile)

```
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$
```

 My Planning(/planning/me)

Repo:


-  Projects(/projects/current)
 - GitHub repository: alx-backend-storage
 - Directory: 0x01-NoSQL
-  File: 4-match
 - QA Reviews I can make(/corrections/to_review)


Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)


5. Count


mandatory


 Score: 100.0% (Checks completed: 100.0%)
Curriculums(/dashboards/my_curriculums)

 Write a script that displays the number of documents in the collection school :
Concepts(/concepts)

- 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\$

 Tools(/dashboards/my_tools)
Repo:

-  GitHub repository: alx-backend-storage
- Video on demand(/dashboards/videos)
- Directory: 0x01-NoSQL
- File: 5-count


Peers(/users/peers)

 Discord(<https://discord.com/app>)

6. Update

mandatory

Score: 100.0% (Checks completed: 100.0%)
My Profile(/users/my_profile)

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



Home(/)

```
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
Projects(/projects/current)
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", "address" : "972 Mission street" }
bye
guillaume@ubuntu:~/0x01$
```



Curriculums(/dashboards/my_curriculums)

Repo:



- Concepts(/concepts)
- GitHub repository: `atx-backend-storage`
- Directory: `0x01-NoSQL`
- File: `6-update`



Conference rooms(/dashboards/video_rooms)



Check submission



Get a sandbox

View results

Servers(/servers)

7. Delete by match

mandatory



Sandboxes(/user_containers/current)

Score: 100.0% (Checks completed: 100.0%)



Tools(/dashboards/my_tools)

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

Video on demand(/dashboards/videos)

```
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
Discord(https://discord.com/app)
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
My Profile(/users/my_profile)
guillaume@ubuntu:~/0x01$
```



Repo:

- (/)
- GitHub repository: alx-backend-storage
- Directory: 0x01-NoSQL
- Home (/)
- File: 7-delete



Check submission



Get a sandbox

View results

9. List all documents in Python

Projects(/projects/current)

mandatory

Score: 100.0% (Checks completed: 100.0%)

QA Reviews I can make(/corrections/to_review)

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

- Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)
- 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:~/dashboards/my8main.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')))
```



```
Sandboxes(/user_containers/current)
guillaume@ubuntu:~/0x01$
guillaume@ubuntu:~/0x01$ ./8-main.py
```



```
[5a8f60cfd4321e1403ba7aba] Holberton school
[5a8f60cfd4321e1403ba7aba] UCSD
guillaume@ubuntu:~/0x01$
```



Video on demand(/dashboards/videos)

Repo:

- GitHub repository: alx-backend-storage
- Peers(/users/peers)
- Directory: 0x01-NoSQL
- File: 8-all.py



Discord(<https://discord.com/app>)

Check submission



Get a sandbox

View results



My Profile(/users/my_profile)

9. Insert a document in Python

mandatory

(/)

Score: 100.0% (Checks completed: 100.0%)



Home(/)

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`



My Planning(/planning/me)



Projects(/projects/current)

guillaume@ubuntu:~/0x01\$ cat 9-main.py

```
#!/usr/bin/env python3
```

```
""" 9-main """
```



QA Reviews I can make(/corrections/to_review)

```
from pymongo import MongoClient
```

```
list_all = __import__('8-all').list_all
```



Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

```
if __name__ == "__main__":
```

```
    client = MongoClient('mongodb://127.0.0.1:27017')
```

```
    school_collection = client.my_db.school
```



Curriculum(/dashboards/my_curriculum)

```
    new_school_id = insert_school(school_collection, name="UCSF", address="505 Pa
```

```
rnassus Ave")
```

```
    print("New school created: {}".format(new_school_id))
```



Concepts(/concepts)

```
    schools = list_all(school_collection)
```

```
    for school in schools:
```



Conference rooms(/dashboards/video_rooms)

```
        print("[{}] {}".format(school.get('_id'), school.get('name'), school.g
```

```
et('address', "")))
```



```
guillaume@ubuntu:~/0x01$
```

Servers(/servers)

```
guillaume@ubuntu:~/0x01$ ./9-main.py
```

```
New school created: 5a8f60cfd4321e1403ba7abb
```



```
[5a8f60cfd4321e1403ba7ab9]
```

Sandboxes(/user_containers/current)

```
[5a8f60cfd4321e1403ba7aba] UCSF
```

```
[5a8f60cfd4321e1403ba7abb] UCSF 505 Parnassus Ave
```



```
guillaume@ubuntu:~/0x01$
```

Tools(/dashboards/my_tools)



Repo: Video on demand(/dashboards/videos)

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



Peers(/users/peers)

Check submission

> Get a sandbox

View results



Discord(<https://discord.com/app>)

10. Change school topics

mandatory

Score: 100.0% (Checks completed: 100.0%)

My Profile(/users/my_profile)

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

```

Home(/)
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
Projects(/projects/current)
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", "Alg
orithm"])
    Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)
    schools = list_all(school_collection)
    for school in schools:
        print("[{}] {} {}".format(school.get('_id'), school.get('name'), school.g
et('topics', "")))
    Curriculums(/dashboards/my_curriculums)
    update_topics(school_collection, "Holberton school", ["iOS"])
    Concepts(/concepts)
    schools = list_all(school_collection)
    for school in schools:
        print("[{}] {} {}".format(school.get('_id'), school.get('name'), school.g
et('topics', "")))
    Conference rooms(/dashboards/video_rooms)

guillaume@ubuntu:~/0x01$
Servers(/servers)
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$

```

Video on demand(/dashboards/videos)

Repo:

- GitHub repository: alx-backend-storage
- Directory: 10-main
- File: 10-update_topics.py

Discord(<https://discord.com/app>)

Check submission

Get a sandbox

View results



11. Where can I learn Python?

mandatory

My Profile(/users/my_profile)
Score: 100.0% (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



Home(/)

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



evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

j_schools = [

{ 'name': "Holberton school", 'topics': ["Algo", "C", "Python", "React"] },

{ 'name': "UCSF", 'topics': ["Algo", "MongoDB"] },

{ 'name': "UCLA", 'topics': ["Algo", "Python"] },

{ 'name': "UCSD", 'topics': ["Cassandra"] },

{ 'name': "Stanford", 'topics': ["C", "React", "Javascript"] }

] Concepts(/concepts)

for j_school in j_schools:

insert_school(school_collection, **j_school)



Conference rooms(/dashboards/video_rooms)

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\$



Video on demand(/dashboards/videos)

Repo:

- GitHub repository: alx-backend-storage



- Directory: 0x01-NoSQL

- File: 11-schools_by_topic.py



Discord(<https://discord.com/app>)

Check submission

> Get a sandbox

View results



12. Log stats

mandatory

Score: 0.0% (0 out of 100 points)

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



- Database: logs
- (/) Collection: nginx

- Display (same as the example):



Home (/)
 • first line: x logs where x is the number of documents in this collection



My Planning (/planning/me)
 • 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)



Projects (/projects/current)
 • one line with the number of documents with:
 ▪ method=GET
 ▪ path=/status



QA Reviews I can make (/corrections/to_review)

You can use this dump as data sample: dump.zip (/token/0szbpsIKvH3RqKb_2HUeoQ)

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



Evaluation quizzes (/dashboards/my_current_evaluation_quizzes)



Curriculums (/dashboards/my_curriculums)



Concepts (/concepts)



Conference rooms (/dashboards/video_rooms)



Servers (/servers)



Sandboxes (/user_containers/current)



Tools (/dashboards/my_tools)



Video on demand (/dashboards/videos)



Peers (/users/peers)



Discord (https://discord.com/app)



My Profile (/users/my_profile)

```

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
2018-02-23T20:12:37.816+0000 preparing collections to restore from
2018-02-23T20:12:37.816+0000 reading metadata for logs.nginx from dump/logs/nginx.metadata.json
2018-02-23T20:12:37.816+0000 reading metadata for logs.nginx from dump/logs/nginx.bson
2018-02-23T20:12:40.804+0000 [##.....] logs.nginx 1.21MB/1
3.4MB (9.0%)
2018-02-23T20:12:42.803+0000 [#####] logs.nginx 2.88MB/1
3.4MB (21.4%)
2018-02-23T20:12:46.803+0000 [#####.....] logs.nginx 4.22MB/1
3.4MB (31.4%)
2018-02-23T20:12:49.803+0000 [#####.....] logs.nginx 5.73MB/1
3.4MB (42.7%)
2018-02-23T20:12:52.803+0000 [#####.....] logs.nginx 7.23MB/1
3.4MB (53.8%)
2018-02-23T20:12:55.803+0000 [#####.....] logs.nginx 8.53MB/1
3.4MB (63.5%)
2018-02-23T20:12:58.803+0000 [#####.....] logs.nginx 10.1MB/1
3.4MB (74.9%)
2018-02-23T20:13:01.803+0000 [#####.....] logs.nginx 11.3MB/1
3.4MB (83.9%)
2018-02-23T20:13:04.803+0000 [#####.....] logs.nginx 12.8MB/1
3.4MB (94.9%)
2018-02-23T20:13:06.228+0000 [#####] logs.nginx 13.4MB/1
3.4MB (100%)
2018-02-23T20:13:06.230+0000 no indexes to restore
2018-02-23T20:13:06.231+0000 finished restoring logs.nginx (94778 documents)
2018-02-23T20:13:06.231+0000 done
guillaume@ubuntu:~/0x01$
guillaume@ubuntu:~/0x01$ ./12-log_stats.py
94778 Videos
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



(/)

Check submission

Mark submission

> Get a sandbox

View results



Home(/)

13. Regex filter

#advanced



Score: 100.0% (Checks completed: 100.0%)



Write a script that lists all documents with name starting by Holberton in the collection school : Projects(/projects/current)

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



guillaume@ubuntu:~\$ cat 100-find.py | mongo my_db

MongoDB shell version v3.6.3

connecting to: mongod://127.0.0.1:27017/my_db



MongoDB shell version v3.6.3

{ "_id" : ObjectId("5a90731fd4321e1e5a3f53e3"), "name" : "Holberton school" }

{ "_id" : ObjectId("5a90731fd4321e1e5a3f53e3"), "name" : "Holberton School" }

{ "_id" : ObjectId("5a90731fd4321e1e5a3f53e3"), "name" : "Holberton-school" }



bye
guillaume@ubuntu:~\$



Concepts(/concepts)

Repo:



- GitHub repository: aly-backend-storage
- Directory: 0x01-NoSQL
- File: 100-find



Servers(/servers)

Check submission

> Get a sandbox

View results



Sandboxes(/user_containers/current)



Tools(/dashboards/my_tools)

14. Top students

#advanced



Video on demand(/dashboards/videos)

Score: 100.0% (Checks completed: 100.0%)

Write a Python function that returns all students sorted by average score:



- Prototype: `def top_students(mongo_collection):`
- `mongo_collection` will be the `pymongo` collection object



- The top must be ordered
- Discord(<https://discord.com/app>)
- The average score must be part of each item returns with key = `averageScore`



My Profile(/users/my_profile)

```

guillaume@ubuntu:~/0x01$ cat 101-main.py
#!/usr/bin/env python3
"""101-main"""
from pymongo import MongoClient
list_all = __import__('8-all').list_all
insert_school = __import__('9-insert_school').insert_school
top_students = __import__('101-students').top_students

if __name__ == '__main__':
    client = MongoClient('mongodb://127.0.0.1:27017')
    students_collection = client.my_db.students
    Projects(/projects/current)
    j_students = [
        { 'name': "John", 'topics': [{ 'title': "Algo", 'score': 10.3 }, { 'title': "C", 'score': 4.9 }, { 'title': "Python", 'score': 7.9 } ] },
        { 'name': "Bob", 'topics': [{ 'title': "Algo", 'score': 5.4 }, { 'title': "C", 'score': 4.9 }, { 'title': "Python", 'score': 7.9 } ] },
        { 'name': "Sonia", 'topics': [{ 'title': "Algo", 'score': 14.8 }, { 'title': "C", 'score': 8.8 }, { 'title': "Python", 'score': 15.7 } ] },
        { 'name': "Amy", 'topics': [{ 'title': "Algo", 'score': 9.1 }, { 'title': "C", 'score': 14.2 }, { 'title': "Python", 'score': 4.8 } ] },
        { 'name': "Julia", 'topics': [{ 'title': "Algo", 'score': 10.5 }, { 'title': "C", 'score': 10.2 }, { 'title': "Python", 'score': 10.1 } ] }
    ]
    for j_student in j_students:
        insert_school(students_collection, **j_student)

    students = list_all(students_collection)
    for student in students:
        print("[{}] {} - {}".format(student.get('_id'), student.get('name'), student.get('topics')))

    top_students = top_students(students_collection)
    for student in top_students:
        print("{} - {}".format(student.get('_id'), student.get('name'), student.get('averageScore')))

guillaume@ubuntu:~/0x01$ ./101-main.py
[5a90776bd4321e1ec94fc408] John - [{ 'title': 'Algo', 'score': 10.3 }, { 'title': 'C', 'score': 4.9 }, { 'title': 'Python', 'score': 7.9 }]
[5a90776bd4321e1ec94fc409] Bob - [{ 'title': 'Algo', 'score': 5.4 }, { 'title': 'C', 'score': 4.9 }, { 'title': 'Python', 'score': 7.9 }]
[5a90776bd4321e1ec94fc40a] Sonia - [{ 'title': 'Algo', 'score': 14.8 }, { 'title': 'C', 'score': 8.8 }, { 'title': 'Python', 'score': 15.7 }]
[5a90776bd4321e1ec94fc40b] Amy - [{ 'title': 'Algo', 'score': 9.1 }, { 'title': 'C', 'score': 14.2 }, { 'title': 'Python', 'score': 4.8 }]
[5a90776bd4321e1ec94fc40c] Julia - [{ 'title': 'Algo', 'score': 10.5 }, { 'title': 'C', 'score': 10.2 }, { 'title': 'Python', 'score': 10.1 }]
[5a90776bd4321e1ec94fc40a] Sonia => 13.1
[5a90776bd4321e1ec94fc40c] Julia => 10.266666666666666
[5a90776bd4321e1ec94fc408] John => 9.533333333333333
[5a90776bd4321e1ec94fc40b] Amy => 9.366666666666665
[5a90776bd4321e1ec94fc409] Bob => 6.066666666666667
guillaume@ubuntu:~/0x01$

```




Repo:

- (/)
- GitHub repository: alx-backend-storage
- Directory: 0x01-NoSQL
- File: 102-log_stats.py



Check submission

My Planning(/planning/me)

> Get a sandbox

View results



15. Log stats - new version

Projects(/projects/current)

#advanced



Score: 0.0% (Checks completed: 0.0%)

QA Reviews I can make(/corrections/to_review)

Improve 12-log_stats.py by adding the top 10 of the most present IPs in the collection nginx of the database logs.

Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

- The IPs top must be sorted (like the example below)



```
guillaume@ubuntu:~/0x01$ ./102-log_stats.py
```

```
94778 Logs(/dashboards/my_curriculums)
```

Methods:

```
method GET: 93842
```

```
method POST: 229
```

```
method PUT: 0
```

```
method PATCH: 0
```

```
method DELETE: 0
```

```
47415 status check
```

IPs:

```
172.31.60.67: 15805
```

```
172.31.2.14: 15805
```

```
172.31.29.194: 15805
```

```
69.162.124.230: 529
```

```
64.124.26.109: 408
```

```
64.62.224.29: 217
```

```
34.207.121.61: 183
```

```
47.88.100.4: 166
```

```
45.249.84.250: 160
```

```
216.244.66.228: 150
```

```
guillaume@ubuntu:~/0x01$
```



Repo:

Peers(/users/peers)

- GitHub repository: alx-backend-storage
- Directory: 0x01-NoSQL
- Discord(<https://discord.com/app>)
- File: 102-log_stats.py



Check submission

Mark submission

> Get a sandbox

View results

My Profile(/users/my_profile)



(/)

Copyright © 2024 ALX, All rights reserved.



Home(/)



My Planning(/planning/me)



Projects(/projects/current)



QA Reviews I can make(/corrections/to_review)



Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)



Curriculums(/dashboards/my_curriculums)



Concepts(/concepts)



Conference rooms(/dashboards/video_rooms)



Servers(/servers)



Sandboxes(/user_containers/current)



Tools(/dashboards/my_tools)



Video on demand(/dashboards/videos)



Peers(/users/peers)



Discord(<https://discord.com/app>)



My Profile(/users/my_profile)