Quest 6. MongoDB

SEUNGKI LEE

Part A. Screen Shot

CODE

```
import pymongo
from pymongo import MongoClient

client = MongoClient('mongodb://seungkil:hu44kWb5@smgo7db01.smu.edu:27017/seungkildb')
db = client.seungkildb

players = [
    {
        "name": "Mesut Ozil",
        "age": 29,
        "position": "AM"
    }, {
        "name": "Aaron Ramsey",
        "age": 27,
        "position": "CM"
    }, {
        "name": "Pierre-Emerick Aubameyang",
        "age": 28,
        "position": "ST"
    }, {
        "name": "Laurent Koscielny",
        "age": 31,
        "position": "CB"
    }, {
        "name": "Alexandre Lacazette",
        "age": 27,
        "position": "ST"
    }
]


for p in players:
        db.player.insert(p)

for p in db.player.find():
        print(p)
```

Genuse DB Call

```
seungkil@genuse26.engr.smu.edu$ nano pymongo_test.py
seungkil@genuse26.engr.smu.edu$ python pymongo_test.py
{u'position': u'AM', u'age': 29, u'_id': ObjectId('5bb22fd1986bd93051f6ae21'), u'name': u'Mesut Ozil'}
{u'position': u'CM', u'age': 27, u'_id': ObjectId('5bb22fd1986bd93051f6ae22'), u'name': u'Aaron Ramsey'}
{u'position': u'ST', u'age': 28, u'_id': ObjectId('5bb22fd1986bd93051f6ae23'), u'name': u'Pierre-Emerick Aubameyang'}
{u'position': u'CB', u'age': 31, u'_id': ObjectId('5bb22fd1986bd93051f6ae24'), u'name': u'Laurent Koscielny'}
{u'position': u'ST', u'age': 27, u'_id': ObjectId('5bb22fd1986bd93051f6ae25'), u'name': u'Alexandre Lacazette'}
```

Part B. MongoDB Exploration Handout

SEUNGKI LEE

```
Category: mongoimport
Action: Load CSV file into mongo
```

Basic Import

```
mongoimport -d <db_name> -c <collection_name> --file <file_name.extension>
```

you could be more explicit

```
mongoimport --db <db_name> --collection <collection_name> --file <file_name.extension>
```

or change the order around, as long as it is not --`file` flag

```
# Order of flags does not matter except --file flag
mongoimport -c <collection_name> -d <db_name> --file <file_name.extension>
```

Merging DB as you import

consider following document

```
{
    "_id":  ObjectId("580100f4da893943d393e909")
    "name": "Jake Carlson",
    "major": "Computer Science"
}
```

Now, let's assume that there is another document in the same file

```
{
    "_id":  ObjectId("580100f4da893943d393e909")
    "email": "jcarlson@ballin'.com",
    "hobby": ["drinking", "ballin'", "shooting nail guns at people"]
}
```

mongoimport has some very cool feature to not only import the documents, but merge the documents with the same ObjectId.

```
mongoimport -c smart -d people --mode merge --file people_SMU_2018.json
```

This command will give you

```
{
    "_id":  ObjectId("580100f4da893943d393e909")
    "name": "Jake Carlson",
    "major": "Computer Science",
    "email": "jcarlson@ballin.com",
    "hobby": [
        "drinking",
        "ballin'",
        "shooting nail guns at people"
        ]
}
```

Importing CSV

Let's assume we have csv data

```
# name      | gender | age | first language | is engaged?
SeungKi Lee,  M,      25,   Korean,          F
Chi Tran,     M,      23,   Vietnamese,      T
Jake Carlson, M,      22,   C++,             F?
```

We could just load the data, but this will most likely load everything as string. That can create a problem when we want to work with the data later on. So we can specify the type of the field as we import.

```
mongoimport -d people -c info --type csv --columnsHaveTypes -f "name.string(), gender.string(),
age.int32(), language.string(), is_engaged.boolean()" --file ~/people_file_2018.csv
```

This will force the type of each field, making it easier to work with!