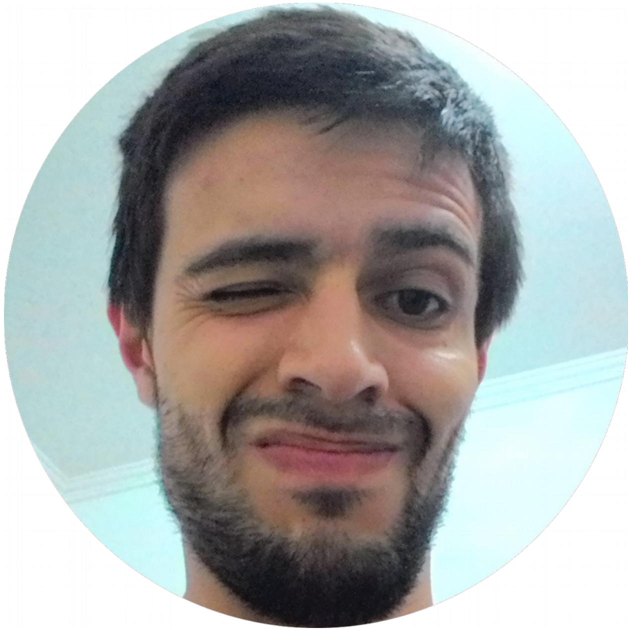
MongoDB CRUD

(Create, Read, Update e Delete)

com Pymongo



ABRA SEU CÓDIGO!

Por um mundo livre e inteligente!!!!!!!!

debian

Linux Professional Institute

Flask

web development, one drop at a time

django

luizalabs

Alguns motivos que me fizeram gostar do MongoDB

- Muito rápido de se desenvolver qualquer coisa!!
  - Linguagem simples!

- Schemaless – ótimo para lidar com dados não estruturados
  - Livros que possuem N autores e autores que possuem N livros
  - Categorias que possuem mais categorias
  - “Nuvem” de tags em tópicos de um blog

- Escalável – ReplicaSets/Shards!

- GridFS – armazene arquivos grandes!

Instalando MongoDB

- Doc oficial:

- **http://docs.mongodb.org/manual/installation/**
- **https://docs.docker.com/examples/mongodb/**

- Minha doc:

```
# docker pull mongo
# docker run -t -i -p 27017:27017 -v /data_mongodb/db:/data/db  mongo /bin/bash
```

- `-t -i`: opções para mostrar o que está rolando
- `-p 27017:27017`: portas local:container
- `-v /data_mongodb/db:/data/db`: diretório "local" : diretório container
- `mongo`: image
- `/bin/bash`

```
# mongod &
```

Instalando Pymongo

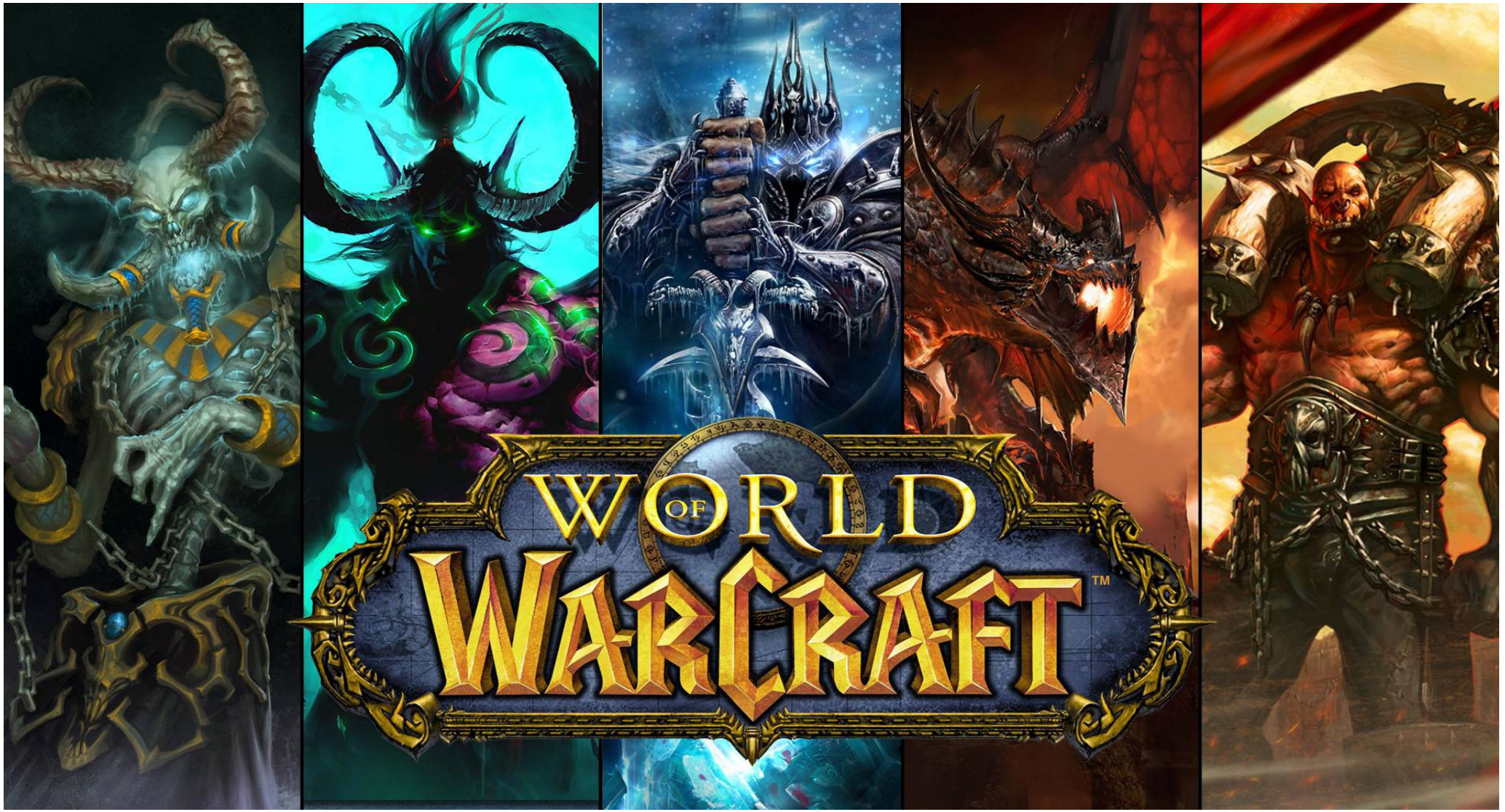
- **Doc oficial:**

- **http://api.mongodb.org/python/current/installation.html**

- **Minha doc:**

```
$ mkvirtualenv -p /usr/bin/python3.4 grupymongo
(grupymongo)$ pip install pymongo
```

Ops... mudando o título da LT...



Como aprender Pymongo no World of Warcraft!??

Faôlin
100 Blood Elf Rogue
The Path of Darkness
Goldrinn

9737 13 7

Summary
Auction Summary
Buy
Sell
Bids (0)
Auctions (0)

Buy

Neutral

Item name: | Categories: All | Level range: | Minimum rarity: Common

All
Weapon
Armor
Container
Consumable
Glyph
Trade Goods
Recipe
Gem
Miscellaneous
Quest
Battle Pets

BROWSE | Reset

Showing **1**–**20** of **200** results | Results per page: 20 | Next

Name / Rarity	Level	Time Left	Bid / Buyout	
Strange Dust Dehlilah	0	Very long	0 0 8 2 16 65	Bid Buyout
Strange Dust Dehlilah	1	0	Very long	0 0 8 2 16 65	Bid Buyout
Strange Dust Dehlilah	1	0	Very long	0 0 8 2 16 65	Bid Buyout
Strange Dust Dehlilah	1	0	Very long	0 0 8 2 16 65	Bid Buyout
Strange Dust Dehlilah	1	0	Very long	0 0 8 2 16 65	Bid Buyout
Strange Dust Dehlilah	1	0	Very long	0 0 8 2 16 65	Bid Buyout

Mas tudo hoje são apis.....

http://us.battle.net/api/wow/auction/data/goldrinn


```
{
   "files": [
      {
         "lastModified": 1440200012000,
         "url": "http://us.battle.net/auction-data/207e97cc61cceeb5bc451a260cf898cd/auctions.json"
      }
   ]
}
```


http://us.battle.net/auction-data/207e97cc61cceeb5bc451a260cf898cd/auctions.json


```
{
   "auctions": [
      {
         "auc": 488323241,
         "bid": 90029,
         "buyout": 94746,
         "context": 0,
         "item": 107528,
         "owner": "Nitkity",
         "ownerRealm": "Goldrinn",
         "quantity": 12,
         "rand": 0,
         "seed": 0,
         "timeLeft": "VERY_LONG"
      },
      {
```


Só não é fácil ler
499380 linhas!!! :-o

insert_one e insert_many

```python
import json
import requests
import pymongo

############### PART 1 - Insert One ########################

# establishing connection
connection = pymongo.MongoClient("mongodb://localhost")

# selecting on database
db = connection.wow

# selecting collection
last_url = db.last_url

# drop collection if exists
last_url.drop()

# get url from server
discover_ah = requests.get(
    "http://us.battle.net/api/wow/auction/data/goldrinn")
# parse the json into python dict
parsed = json.loads(discover_ah.content.decode('utf-8'))

# Insert one \o/
result = last_url.insert_one(parsed)
print("Inserted id: ", result.inserted_id)

```

insert_one e insert_many

```
29 ############## PART 2 - Insert Many ######################
30
31 # selecting collection
32 ah_items = db.ah_items
33
34 # drop collection if exists
35 ah_items.drop()
36
37 # get url from server
38 ah_url = parsed.get("files")[0].get("url")
39 all_items_ah = requests.get(ah_url)
40
41 # parse the json into python dict
42 parsed = json.loads(all_items_ah.content.decode('utf-8'))
43
44 # qty of auctions
45 print("Qty Auctions: ", len(parsed.get("auctions")))
46
47 from time import time
48 start = time()
49 # Insert many \o/
50 result = ah_items.insert_many(parsed.get("auctions"))
51 print("Finish: ", time() - start)
52 print("Inserted {} new ids".format(len(result.inserted_ids)))
53
```

insert_one e insert_many

```
29 ############## PART 2 - Insert Many ######################
30
31 # selecting collection
32 ah_items = db.ah_items
33
34 # drop collection if exists
35 ah_items.drop()
36
37 # get url from server
38 ah_url = parsed.get("files")[0].get("url")
39 all_items_ah = requests.get(ah_url)
40
41 # parse the json into python dict
42 parsed = json.loads(all_items_ah.content.decode('utf-8'))
43
44 # qty of auctions
45 print("Qty Auctions: ", len(parsed.get("auctions")))
46
47 from time import time
48 start = time()
49 # Insert many \o/
50 result = ah_items.insert_many(parsed.get("auctions"))
51 print("Finish: ", time() - start)
52 print("Inserted {} new ids".format(len(result.inserted_ids)))
53
```

 # Saída do script

$ python inserts.py
Inserted id: 55d67d9576467c3b8663162d
Qty Auctions: 41143
Finish: 0.634172677993744
Inserted 41143 new ids

fine_one e find

```
1  import pymongo
2  good_gems = [76692, 76693, 76694, 76695, 76696, 76636, 76637, 76638, 76639,
3               76697, 76698, 76699, 76700, 76701, 76680, 76681, 76682, 76683,
4               76684, 76685, 76686, 76687, 76688, 76689, 76690, 76691, 89674,
5               89680, 76640, 76641, 76642, 76643, 76644, 76645, 76646, 76647,
6               76648, 76649, 76650, 76651, 76652, 76653, 76654, 76655, 76656,
7               76657, 93705, 76658, 76659, 76660, 76661, 76662, 76663, 76664]
8
9  # establishing connection
10 connection = pymongo.MongoClient("mongodb://localhost")
11 # selecting on database
12 db = connection.wow
13 # selecting collection
14 ah_items = db.ah_items
15
16 query = {'item': {"$in": good_gems}}
17 doc = ah_items.find_one(query)  # ah.items.find_one({'item': 76692})
18 print(doc)
19
20 from time import time
21 start = time()
22 cursor = ah_items.find(query)
23 print("Finish: ", time() - start)
24
25 for doc in cursor:
26     print("Owner {} Buyout {} Item {}".format(
27         doc.get('owner'), doc.get('buyout'), doc.get('item')))
28
```

fine_one e find

```
import pymongo
good_gems = [76692, 76693, 76694, 76695, 7
             76697, 76698, 76699, 76700, 7
             76684, 76685, 76686, 76687, 7
             89680, 76640, 76641, 76642, 7
             76648, 76649, 76650, 76651, 7
             76657, 93705, 76658, 76659, 7

# establishing connection
connection = pymongo.MongoClient("mongodb:
# selecting on database
db = connection.wow
# selecting collection
ah_items = db.ah_items

query = {'item': {"$in": good_gems}}
doc = ah_items.find_one(query)  # ah.items
print(doc)

from time import time
start = time()
cursor = ah_items.find(query)
print("Finish: ", time() - start)

for doc in cursor:
    print("Owner {} Buyout {} Item {}".for
        doc.get('owner'), doc.get('buyout'
```

 # Saída do script com 152 gems

$ python find_gems.py

```
{'bid': 485450, 'rand': 0, 'timeLeft': 'VERY_LONG', '_id': ObjectId('55d67d9976467c3b866316c8'), 'auc': 486939890, 'buyout': 511000, 'context': 0, 'owner': 'Manndrake', 'ownerRealm': 'Goldrinn', 'seed': 1044495360, 'quantity': 1, 'item': 76694}

Finish:  2.0742416381835938e-05

Owner  Manndrake
Buyout  511000
Item  76694
-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~
Owner  Ogartitopulo
Buyout  500000
Item  76689
-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~
Owner  Oldmate
Buyout  970900
Item  76694
-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~
```

update_many

```
import pymongo
good_gems = [76692, 76693, 76694, 76695, 76696, 76636, 76637, 76638, 76639,
             76697, 76698, 76699, 76700, 76701, 76680, 76681, 76682, 76683,
             76684, 76685, 76686, 76687, 76688, 76689, 76690, 76691, 89674,
             89680, 76640, 76641, 76642, 76643, 76644, 76645, 76646, 76647,
             76648, 76649, 76650, 76651, 76652, 76653, 76654, 76655, 76656,
             76657, 93705, 76658, 76659, 76660, 76661, 76662, 76663, 76664]

# establishing connection
connection = pymongo.MongoClient("mongodb://localhost")
# selecting on database
db = connection.wow
# selecting collection
ah_items = db.ah_items

# update all the docs
from time import time
start = time()
result = ah_items.update_many(
    {'item': {"$in": good_gems}},
    {'$set': {'good_gem': True}}
)
print("Finish: ", time() - start)
print("Matched count: ", result.matched_count)
print("Modified count: ", result.modified_count)
```

update_many

```python
import pymongo
good_gems = [76692, 76693, 76694, 76695, 76696, 76636, 76637, 76638, 76639,
             76697, 76698, 76699, 76700,
             76684, 76685, 76686, 76687,
             89680, 76640, 76641, 76642,
             76648, 76649, 76650, 76651,
             76657, 93705, 76658, 76659,

# establishing connection
connection = pymongo.MongoClient("mongod
# selecting on database
db = connection.wow
# selecting collection
ah_items = db.ah_items

# update all the docs
from time import time
start = time()
result = ah_items.update_many(
    {'item': {"$in": good_gems}},
    {'$set': {'good_gem': True}}
)
print("Finish: ", time() - start)
print("Matched count: ", result.matched_count)
print("Modified count: ", result.modified_count)
```

 # Saída do script com 152 gems

$ python update_gems.py
Finish: 0.02341437339782715
Matched count: 87
Modified count: 87

Console mongo/pymongo

```python
import pymongo
# establishing connection
connection = pymongo.MongoClient("mongodb://localhost")
# selecting on database
db = connection.wow
# selecting collection
ah_items = db.ah_items
import ipdb
ipdb.set_trace()

# On Python:
# ipdb> ah_items.count()
# 41143
# ipdb> ah_items.count({'good_gem': True})
# 87
# ipdb> ah_items.count({'good_gem': {"$ne": True}})
# 41056

# On MongoDB:
# > db.ah_items.count()
# 41143
# > db.ah_items.find({good_gem: true}).count()
# 87
# > db.ah_items.find({good_gem: {$ne: true} }).count()
# 41056

```

delete_many

```
1  import pymongo
2
3  # establishing connection
4  connection = pymongo.MongoClient("mongodb://localhost")
5  # selecting on database
6  db = connection.wow
7  # selecting collection
8  ah_items = db.ah_items
9
10 # update all the docs
11 from time import time
12 start = time()
13 result = ah_items.delete_many({'good_gem': {"$ne": True}})
14 print("Finish: ", time() - start)
15 print("Deleted count: ", result.deleted_count)
16
```

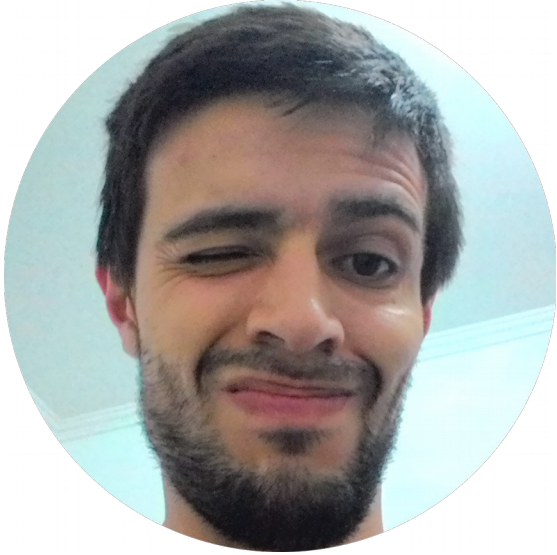
delete_many

```python
import pymongo

# establishing connection
connection = pymongo.MongoClient("mongodb:/
# selecting on database
db = connection.wow
# selecting collection
ah_items = db.ah_items

# update all the docs
from time import time
start = time()
result = ah_items.delete_many({'good_gem': {"$ne": True}})
print("Finish: ", time() - start)
print("Deleted count: ", result.deleted_count)

```

 # Saída do script

$ python delete_many.py
Finish: 0.17556357383728027
Deleted count: 41056

OBRIGADO!



Referências:

Tutorial Pymongo:
http://api.mongodb.org/python/current/tutorial.html

Curso de MongoDB DE GRAÇA! Usando Python!
https://university.mongodb.com/courses/M101P/about

API World of Warcraft:
https://dev.battle.net/io-docs

Contato:
http://abraseucodigo.com.br