MongoDB and Python

Difficulty Level : Hard   ●   Last Updated : 20 Apr, 2018

Prerequisite : MongoDB : An introduction
MongoDB is a cross-platform, document-oriented database that works on the concept of collections and documents. MongoDB offers high speed, high availability, and high scalability.
The next question which arises in the mind of the people is "Why MongoDB"?
Reasons to opt for MongoDB :

1. It supports hierarchical data structure (Please refer docs for details)
2. It supports associate arrays like Dictionaries in Python.
3. Built-in Python drivers to connect python-application with Database. Example-PyMongo
4. It is designed for Big Data.
5. Deployment of MongoDB is very easy.

MongoDB vs RDBMS

RDBMS	MongoDB
Table	Collection
Row	JSON Document
Index	Index
Join	Embedding & Linking
Partition	Shard

1. First start MongoDB from command prompt using :

Method 1:

```
mongod
```

or

Method 2:

```
net start MongoDB
```


See port number by default is set 27017 (last line in above image).
Python has a native library for MongoDB. The name of the available library is
"PyMongo". To import this, execute the following command:

2. **Create a connection :** The very first after importing the module is to create a MongoClient.

```
from pymongo import MongoClient
client = MongoClient()
```

After this, connect to the default host and port. Connection to the host and port is done explicitly. The following command is used to connect the MongoClient on the localhost which runs on port number 27017.

```
client = MongoClient('host', port_number)
example:- client = MongoClient('localhost', 27017)
```

It can also be done using the following command:

```
client = MongoClient("mongodb://localhost:27017/")
```

3. **Access DataBase Objects :** To create a database or switch to an existing database we use:

   Method 1 : Dictionary-style

```
mydatabase = client['name_of_the_database']
```

   Method2 :

```
mydatabase = client.name_of_the_database
```

   If there is no previously created database with this name, MongoDB will implicitly create one for the user.
   Note : The name of the database fill won't tolerate any dash (-) used in it. The names like my-Table will raise an error. So, underscore are permitted to use in the name.

4. **Accessing the Collection :** Collections are equivalent to Tables in RDBMS. We access a collection in PyMongo in the same way as we access the Tables in the RDBMS. To access the table, say table name "myTable" of the database, say "mydatabase".

Method 2 :

```
mycollection = mydatabase.myTable
```

>MongoDB store the database in the form of dictionaries as shown:>
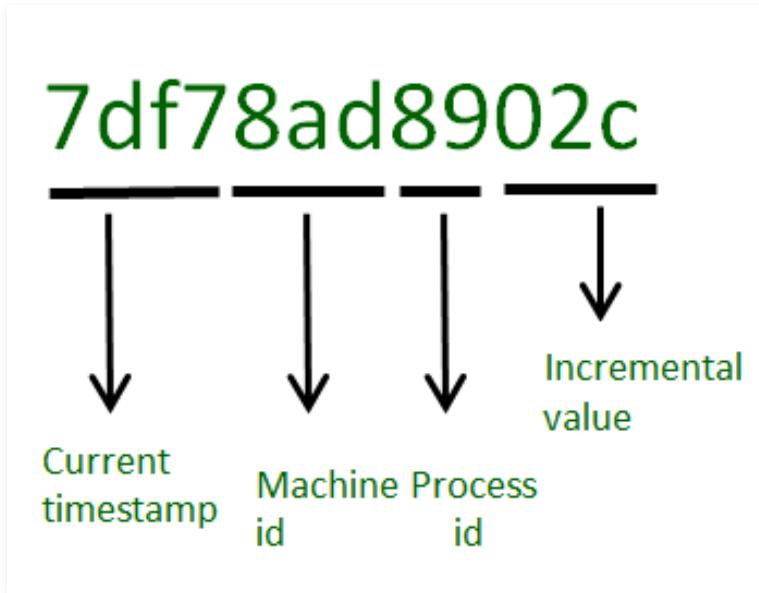
```
record = {
title: 'MongoDB and Python',
description: 'MongoDB is no SQL database',
tags: ['mongodb', 'database', 'NoSQL'],
viewers: 104
}
```

'_id' is the special key which get automatically added if the programmer forgets to add explicitly. _id is the 12 bytes hexadecimal number which assures the

uniqueness of every inserted document.

5. **Insert the data inside a collection :**

Methods used:

```
insert_one() or insert_many()
```

We normally use insert_one() method document into our collections. Say, we wish to enter the data named as record into the 'myTable' of 'mydatabase'.

```
rec = myTable.insert_one(record)
```

The whole code looks likes this when needs to be implemented.

```python
# importing module
from pymongo import MongoClient

# creation of MongoClient
client=MongoClient()

# Connect with the portnumber and host
client = MongoClient("mongodb://localhost:27017/")

# Access database
mydatabase = client['name_of_the_database']

# Access collection of the database
mycollection=mydatabase['myTable']

# dictionary to be added in the database
```

```
viewers: 104
}

# inserting the data in the database
rec = mydatabase.myTable.insert(record)
```

6. **Querying in MongoDB :** There are certain query functions which are used to filer the data in the database. The two most commonly used functions are:

  1. find()

    find() is used to get more than one single document as a result of query.

```
for i in mydatabase.myTable.find({title: 'MongoDB and Python'})
    print(i)
```

    This will output all the documents in the myTable of mydatabase whose title is 'MongoDB and Python'.

  2. **count()**

    count() is used to get the numbers of documents with the name as passed int he parameters.

```
print(mydatabase.myTable.count({title: 'MongoDB and Python'}))
```

    This will output the numbers of documents in the myTable of mydatabase whose title is 'MongoDB and Python'.

    These two query functions can be summed to give a give the most filtered result as shown below.

```
print(mydatabase.myTable.find({title: 'MongoDB and Python'}).count())
```

  3. **To print all the documents/entries inside 'myTable' of database 'mydatabase' :**
    Use the following code:

```
from pymongo import MongoClient

try:
    conn = MongoClient()
    print("Connected successfully!!!")
except:
    print("Could not connect to MongoDB")
```

```
collection = db.myTable

# To find() all the entries inside collection name 'myTable'
cursor = collection.find()
for record in cursor:
    print(record)
```

This article is contributed by Rishabh Bansal and Shaurya Uppal.

If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

 Attention geek! Strengthen your foundations with the **Python Programming Foundation** Course and learn the basics.

To begin with, your interview preparations Enhance your Data Structures concepts with the **Python DS** Course. And to begin with your Machine Learning Journey, join the Machine Learning – Basic Level Course

Previous   Next

RECOMMENDED ARTICLES

Page : **1** 2 3

01 MongoDB Python | Insert and Update Data
09, Nov 17

02 MongoDB Python - Insert and Replace Operations
13, Nov 17

03 MongoDB python | Delete Data and Drop Collection
17, Nov 17

04 Web Scraping CryptoCurrency price and storing it in MongoDB using Python
25, Sep 20

05 MongoDB AND operator ($and)
09, Apr 20

06 Installing MongoDB on Windows with Python
01, Nov 17

07 Python MongoDB - Find
28, Feb 20

08 Python MongoDB - Sort
26, Feb 20

Article Contributed By :

GeeksforGeeks

Vote for difficulty

Current difficulty : Hard

Article Tags :   MongoDB,   GBlog,   Python,   Technical Scripter

Improve Article   Report Issue

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

⠑⠑ GeeksforGeeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Company

About Us

Careers

Privacy Policy

Contact Us

Copyright Policy

Learn

Algorithms

Data Structures

Languages

CS Subjects

Video Tutorials

Practice

Courses

Company-wise

Topic-wise

How to begin?

Contribute

Write an Article

Write Interview Experience

Internships

Videos