Getting Started with MongoDB

Estimated time needed: **30** minutes

Objectives

After completing this lab you will be able to:

- Access the MongoDB server using the command-line interface
- Describe the process of listing and creating collections, which contain documents, and databases, which contain one or more collections
- Perform basic operations on a collection such as inserting, counting and listing documents

About Skills Network Cloud IDE

Skills Network Cloud IDE (based on Theia and Docker) provides an environment for hands on labs for course and project related labs. Theia is an open source IDE (Integrated Development Environment), that can be run on desktop or on the cloud. to complete this lab, we will be using the Cloud IDE based on Theia and MongoDB provided by Skills Network.

Important Notice about this lab environment

Please be aware that sessions for this lab environment are not persisted. Every time you connect to this lab, a new environment is created for you. Any data you may have saved in the earlier session would get lost. Plan to complete these labs in a single session, to avoid losing your data.

Exercise 1 - Start mongodb server

Open the MongoDB database page by clicking the button below:

Open MongoDB Page in IDE

On that page, click the `Create` button to create a MongoDB database.

Exercise 2 - Connect to mongodb server

After creation has finished, click the `MongoDB CLI` button (below the `Create` button from the previous step) to connect to the database using the mongosh CLI.

You should now get connected to the mongodb database, and see an output as in the figure below.

Exercise 3 - Find the version of the server

On the mongo client run the below command.

```
db.version()
```

This will show the version of the mongodb server.

Exercise 4 - List databases

On the mongo client run the below command.

```
show dbs
```

This will print a list of the databases present on the server.

Exercise 5 - Create database

On the mongo client run the below command.

```
use training
```

This will create a new database named **training**. If a database named training already exists, it will start using it.

Exercise 6 - Create collecton

On the mongo client run the below command.

```
db.createCollection("mycollection")
```

This will create a collection name **mycollection** inside the **training** database.

Exercise 7 - List collections

On the mongo client run the below command.

```
show collections
```

This will print the list of collections in your current database.

Exercise 8 - Insert documents into a collection

On the mongo client run the below command.

```
db.mycollection.insert({"color":"white","example":"milk"})
```

The above command inserts the json document {“color”:”white”,”example”:”milk”} into the collection.

Let us insert one more document.

```
db.mycollection.insert({"color":"blue","example":"sky"})
```

The above command inserts the json document {“color”:”blue”,”example”:”sky”} into the collection.

Insert 3 more documents of your choice.

Exercise 9 - Count the number of documents in a collection

On the mongo client run the below command.

```
db.mycollection.countDocuments()
```

This command gives you the number of documents in the collection.

Exercise 10 - List all documents in a collection

On the mongo client run the below command.

```
db.mycollection.find()
```

This command lists all the documents in the collection **mycollection**

Notice that mongodb automatically adds an ‘_id’ field to every document in order to uniquely identify the document.

Exercise 11 - Disconnect from mongodb server

On the mongo client run the below command.

```
exit
```

Practice exercises

1. Problem:

    List databases.

    ▼ Click here for Hint

    Use the ‘show’ command with dbs option.

    ▼ Click here for Solution

    ```
    show dbs
    ```

2. Problem:

    *Create a database named **mydatabase**.*

▼ Click here for Hint

Use the ‘use’ command with the database name.

▼ Click here for Solution

```
use mydatabase
```

3. Problem:

    *Create a collection named **landmarks** in the database **mydatabase**.*

▼ Click here for Hint

Use the ‘createCollection’ command.

▼ Click here for Solution

```
db.createCollection("landmarks")
```

4. Problem:

    List collections

▼ Click here for Hint

Use the ‘show’ command with collections_ option.

▼ Click here for Solution

```
show collections
```

5. Problem:

    Insert details of five landmarks including name, city, and country. Example: Eiffel Tower, Paris, France.

▼ Click here for Hint

Use the ‘db.collection.insert()’ command with the correct options.

▼ Click here for Solution

```
db.landmarks.insert({"name":"Statue of Liberty","city":"New York","country":"USA"})
db.landmarks.insert({"name":"Big Ben","city":"London","country":"UK"})
db.landmarks.insert({"name":"Taj Mahal","city":"Agra","country":"India"})
db.landmarks.insert({"name":"Pyramids","country":"Egypt"})
db.landmarks.insert({"name":"Great Wall of China","country":"China"})
```

6. Problem:

Count the number of documents you have inserted.

▼ Click here for Hint

Use the ‘count’ command on your collections.

▼ Click here for Solution

```
db.landmarks.countDocuments()
```

7. Problem:

List the documents.

▼ Click here for Hint

Use the ‘db.collection.find()’ command.

▼ Click here for Solution

```
db.landmarks.find()
```

8. Problem:

Disconnect from the server.

▼ Click here for Hint

Use the ‘exit’ command.

▼ Click here for Solution

```
exit
```

Authors

Ramesh Sannareddy

Other Contributors

Rav Ahuja