Lab

Topic: Introduction to MongoDB.

Purpose: What is MongoDB. Getting started with MongoDB. Compass graphical client.

Progress

What is MongoDB?

MongoDB represents the currently most popular document-oriented database management system. According to various estimates, it is among the ten most used databases in the world.

Documents instead of strings

Where relational databases store strings, MongoDB stores documents. Unlike strings, documents can store complex information. A document can be submitted as a key-value store.

Key represents a simple label with which a particular piece of data is associated.

However, despite all the differences, there is one feature that brings MongoDB and relational databases closer together. In relational DBMS, there is such a concept as a primary key. This concept describes a column that has unique values. In MongoDB, each document has a unique identifier called _id. And if you don't explicitly specify its value, then MongoDB will automatically generate a value for it.

Each key is compared to a certain value. But one feature should also be taken into account here: if there is a clearly defined structure in relational databases, where there are fields, and if some field has no value, it (depending on the settings of a particular database) can be given the value NULL. MongoDB is different. If a key is not assigned a value, this key is easily omitted from the document and used.

Collections

If the traditional SQL world has tables, the MongoDB world has collections. And if in relational databases the tables store rigidly structured objects of the same type, then the collection can contain different objects that have a different structure and a different set of properties.

Replication

An entire MongoDB system can represent more than just one database hosted on a single physical server. The functionality of MongoDB allows you to host multiple databases on multiple physical servers, and these databases can easily share data and maintain integrity.

The MongoDB data storage system represents a set of replicas. This set has a primary node and may have a set of secondary nodes. All secondary nodes maintain integrity and are automatically updated when the master node is updated. And if the main node fails for some reason, then one of the secondary nodes becomes the main one.

Data format in MongoDB

One of the popular standards for data exchange and data storage is JSON (JavaScript Object Notation). JSON effectively describes complex data structures. The way MongoDB stores data is similar to JSON in this regard, although JSON is not formally used. A format called BSON or short for binary JSON is used to store MongoDB.

BSON allows you to work with data faster: searches and processing are performed faster. Although it should be noted that BSON, unlike storing data in JSON format, has a small disadvantage: in general, data in JSON format takes up less space than in BSON format, on the other hand, this disadvantage is more than compensated by speed.

Cross-platform

MongoDB is written in C++, so it is easy to port to different platforms. MongoDB can be deployed on Windows, Linux, MacOS, Solaris platforms. You can also download the source code and compile MongoDB yourself, but using offsite libraries is recommended.

Ease of use

The absence of a rigid database scheme and, in this regard, the need to recreate this scheme at the slightest change in the concept of data storage make it much easier to work with MongoDB databases and their subsequent scaling. In addition, developers' time is saved. They no longer need to think about rebuilding the database and spend time building complex queries.

But, even considering all the disadvantages of traditional databases and the advantages of MongoDB, it is important to understand that the tasks are different and the methods of solving them are different. In some situations, MongoDB will really improve the performance of your application, for example, if you need to store complex data structures. In another situation, it will be better to use traditional relational databases. Alternatively, you can use a mixed approach: store one type of data in MongoDB and another type of data in traditional databases.

GridFS

One of the challenges when working with any database systems is storing large data. You can store data in files using different programming languages. Some DBMS offer special data types for storing binary DB data (for example, BLOB in MySQL). Unlike relational DBMS, MongoDB allows you to store different documents with different sets of data, but the document size is limited to 16 MB. But MongoDB offers a solution - a special GridFS technology that allows you to store data larger than 16 MB.

The GridFS system consists of two collections. The first collection, called files, stores file names as well as their metadata, such as size. And in another collection, which is called chunks, file data is stored in the form of small segments, usually in segments of 256 KB.

To test GridFS, you can use the special mongofiles utility, which is included in the mongodb package.

Installing and getting started with MongoDB

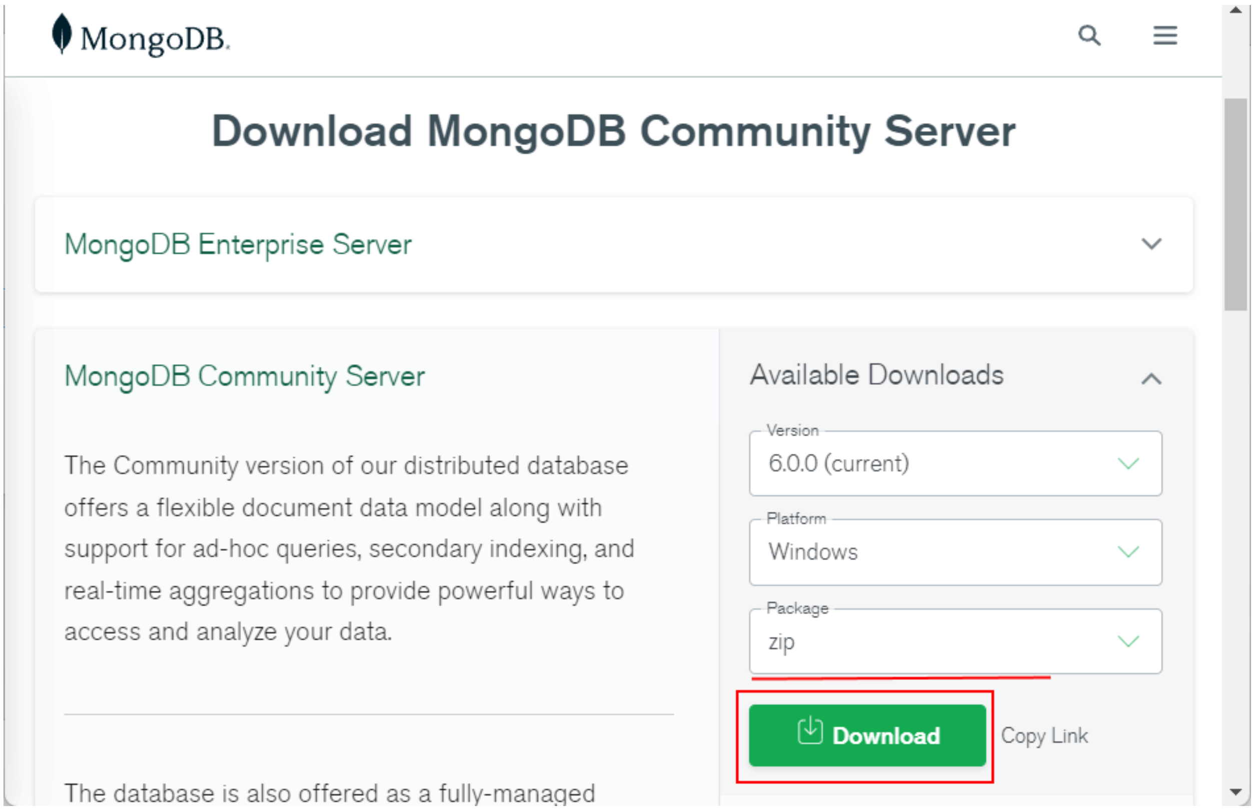
Installing the Mongod server

The official site offers distribution packages for various platforms: Windows, Linux, MacOS, Solaris. And each platform has several distributions. Moreover, there are two types of servers - free Community and paid Enterprise. In this tutorial we will be using the free Community version.

To install MongoDB, download one package from the official websitehttps://www.mongodb.com/try/download/community.

To download all the necessary files, select the required operating system and the appropriate package type. Let's consider the example of installation on the Windows OS.

MongoDB can be downloaded in a number of variants. Yes, there is an msi installer download available for Windows and a zip package download is also available. In fact, it is enough for us to download the zip archive and unpack it in the folder we need. Therefore, we will choose this download option:



Download MongoDB Community Server

MongoDB Enterprise Server

MongoDB Community Server

The Community version of our distributed database offers a flexible document data model along with support for ad-hoc queries, secondary indexing, and real-time aggregations to provide powerful ways to access and analyze your data.

The database is also offered as a fully-managed

Available Downloads

Version: 6.0.0 (current)

Platform: Windows

Package: zip

After downloading the archive package, unzip it to the C:\mongodb folder.

If, after installation, we open the bin folder in the unzipped archive (C:\mongodb\bin), we can find there a bunch of applications that perform a certain role. Let's briefly consider them.

- **mongod**: MongoDB database server. It processes queries, manages the data format, and performs various background database management operations
- **mongoose**: A MongoDB routing service that helps process queries and locate data in a MongoDB cluster

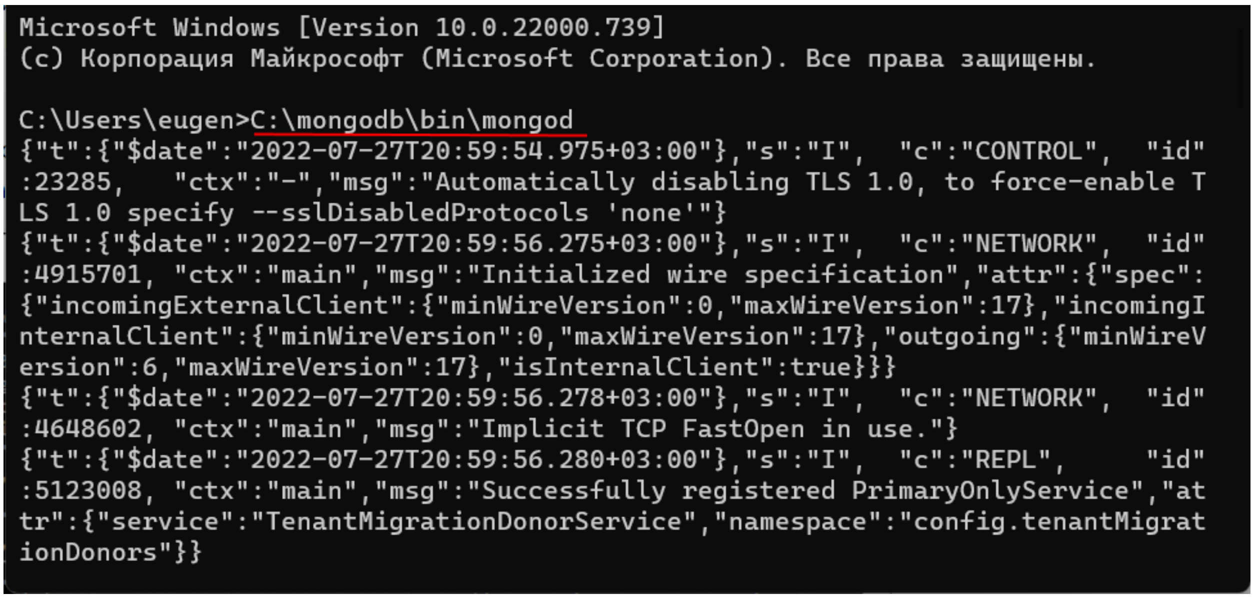
Creating a database directory and starting MongoDB

After installation, you need to create a directory on your hard drive that will contain the MongoDB databases.

On Windows, by default MongoDB stores its databases in the path C:\data\db , so if you're using Windows, you'll need to create a directory accordingly.

If there is a need to use some other file path, it can be passed when starting MongoDB in the --dbpath flag.

So, after creating the DB storage directory, you can start the MongoDB server. The server presents the mongod application, which is located in the bin directory in the server folder. To do this, we will launch the terminal/command line and enter the appropriate commands there. For Windows it will look like this:



```
Microsoft Windows [Version 10.0.22000.739]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\eugen>C:\mongodb\bin\mongod
{"t":{"$date":"2022-07-27T20:59:54.975+03:00"},"s":"I",  "c":"CONTROL",  "id"
:23285,   "ctx":"-","msg":"Automatically disabling TLS 1.0, to force-enable T
LS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2022-07-27T20:59:56.275+03:00"},"s":"I",  "c":"NETWORK",  "id"
:4915701, "ctx":"main","msg":"Initialized wire specification","attr":{"spec":
{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingI
nternalClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireV
ersion":6,"maxWireVersion":17},"isInternalClient":true}}}
{"t":{"$date":"2022-07-27T20:59:56.278+03:00"},"s":"I",  "c":"NETWORK",  "id"
:4648602, "ctx":"main","msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2022-07-27T20:59:56.280+03:00"},"s":"I",  "c":"REPL",     "id"
:5123008, "ctx":"main","msg":"Successfully registered PrimaryOnlyService","at
tr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrat
ionDonors"}}
```

The command prompt will display some service information, such as that the server is running on localhost on port 27017.

And after the successful launch of the server, we will be able to perform database operations through the client.

Installing the Mongosh client

Above we installed the MongoDb server. However, we need a client to work with the server. The simplest client in this case is the MongoDB Shell or mongosh - a console shell for sending requests to the server, which is also provided directly by MongoDB.

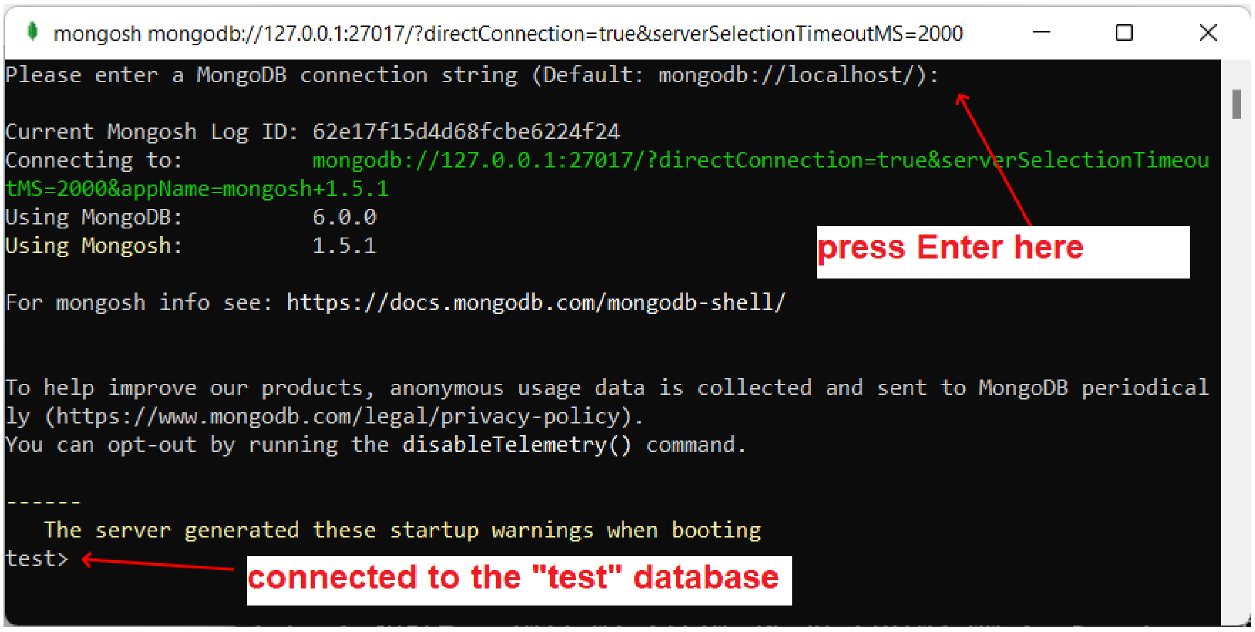
Here again we can choose client versions for different operating systems. For Windows, the client package is available as an msi installer and a zip archive. In this case, we will select the zip archive.

Unzip the downloaded package to the C: Mongosh folder.

If we go to the bin folder in the unpacked archive (that is, C: \ mongosh \ bin), we will find the mongosh console utility there, which will be used to work with the MongoDB server.

Connecting to the server from the client

We use the mongosh client program installed above to interact with the mongodb server. (When working with mongosh, do not forget that we must have the mongod server running). So, let's run the mongosh file, which is located in the installation folder discussed above:


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
press Enter here
connected to the "test" database

```
Please enter a MongoDB connection string (Default: mongodb://localhost/):

Current Mongosh Log ID: 62e17f15d4d68fcbe6224f24
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeou
tMS=2000&appName=mongosh+1.5.1
Using MongoDB:          6.0.0
Using Mongosh:          1.5.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/


To help improve our products, anonymous usage data is collected and sent to MongoDB periodical
ly (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

------
   The server generated these startup warnings when booting
test>
```

When mongosh starts, it will first ask the user which connection string will be used to connect to the MongoDB server. At this point, simply press Enter to use the default MongoDB connection string. And by default, the mongodb server starts on port 27017, and the full connection string looks like this: mongodb://localhost:27017 or mongodb://127.0.0.1:27017

Once connected, the console will display a series of service information and connect to the test database.

Now let's do some of the simplest actions. Enter the following commands in the console in sequence and press Enter after each command:

```
db.users.insertOne( { name: "Tom" } )<font></font>
db.users.find()<font></font>
```

The first use test command sets the test database to be used. Even if there is no such database, it is created automatically.

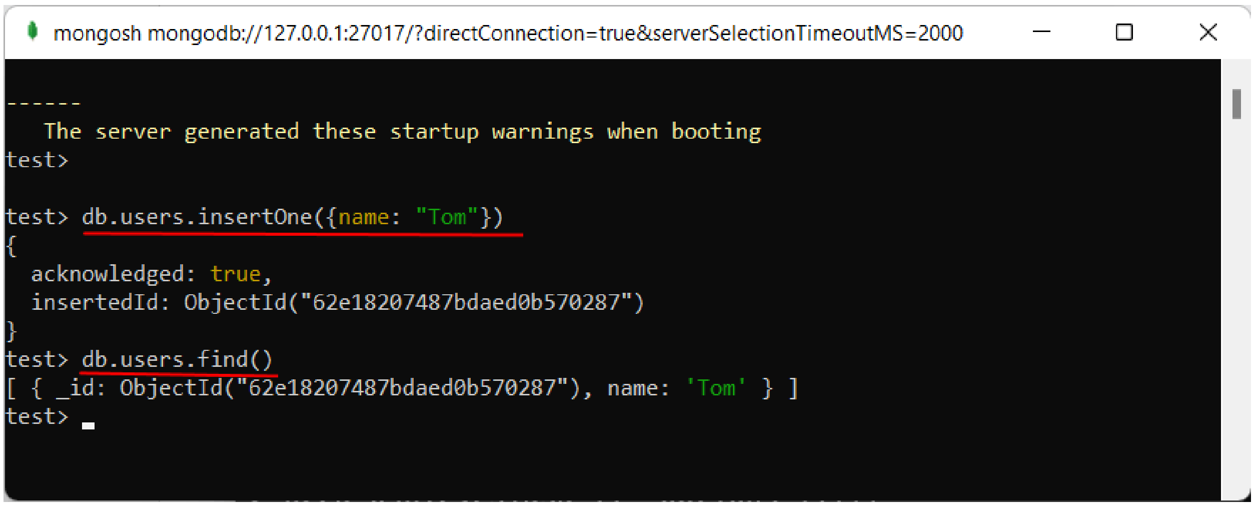
Using the db.users.insertOne() method, an object { name: "Tom" } is added to the users collection of the test database. The db identifier is the current database. In our case, we are connected to the database by default - that is, to the test database, accordingly db here represents the test database. At the same time, it is not important whether such a database exists - if it does not exist, then it is created

After db comes users - this is a collection to which we then add a new object. If SQL we need to create tables in advance, then MongoDB creates collections independently in its absence.

The attached object description is defined in a format you may be familiar with if you've dealt with JSON. That is, in this case, the object has one key "name", which is mapped to the value "Tom". That is, we add a user named Tom.

If the object was successfully added, the console will display the result of the operation, in particular, the identifier of the added object.

And the third command db.users.find() displays all objects from the test database.


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

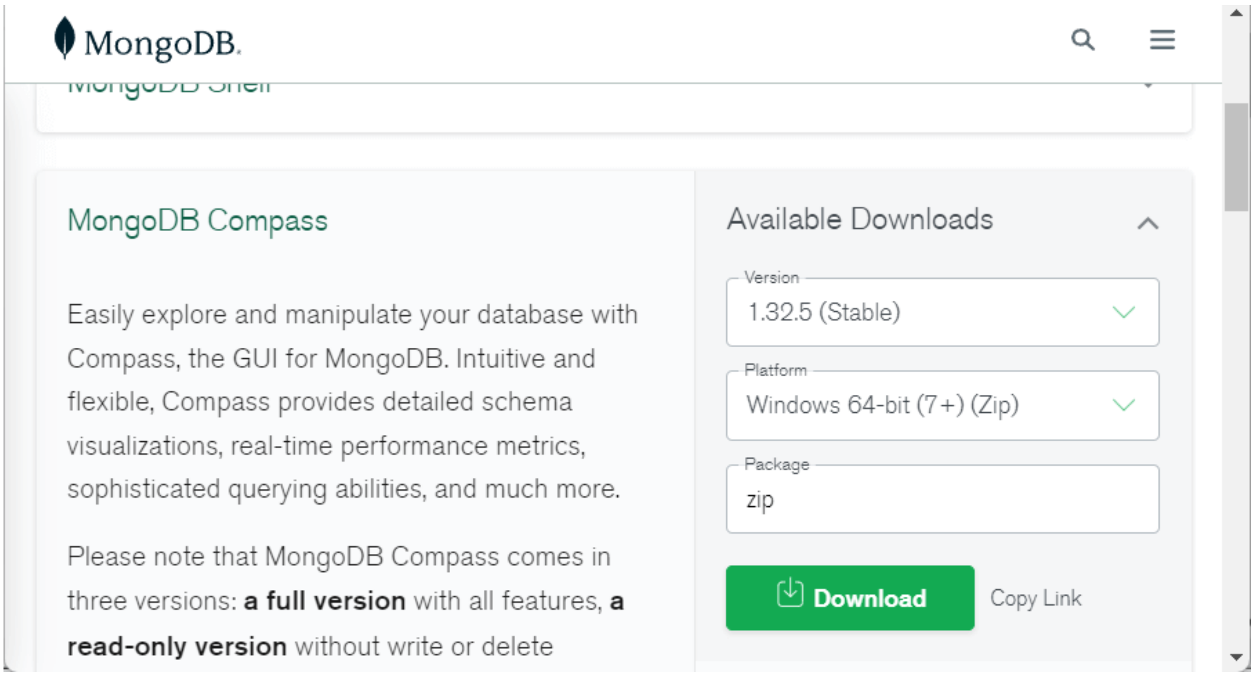
```
------
   The server generated these startup warnings when booting
test>

test> db.users.insertOne({name: "Tom"})
{
  acknowledged: true,
  insertedId: ObjectId("62e18207487bdaed0b570287")
}
test> db.users.find()
[ { _id: ObjectId("62e18207487bdaed0b570287"), name: 'Tom' } ]
test> 
```

From the output, you can see that some obscure ObjectId field has been added to the original object values. As you remember, MongoDB uses the _id field as a unique document identifier. And in this case, ObjectId does represent the value for the _id identifier.

Compass graphical client

You can also use the official MongoDB Compass graphical client to work with MongoDB. To download it, go to the address https://www.mongodb.com/try/download/compass. On this page we can select the download options - the version of Compass and the target operating system. Let's consider the example of installation on Windows.

For Windows, we have two options: download as an msi installer and download a zip archive. In fact, we just need to download the archive and unpack it in the right place. So let's choose this option:



MongoDB Compass

Easily explore and manipulate your database with Compass, the GUI for MongoDB. Intuitive and flexible, Compass provides detailed schema visualizations, real-time performance metrics, sophisticated querying abilities, and much more.

Please note that MongoDB Compass comes in three versions: **a full version** with all features, **a read-only version** without write or delete

Available Downloads

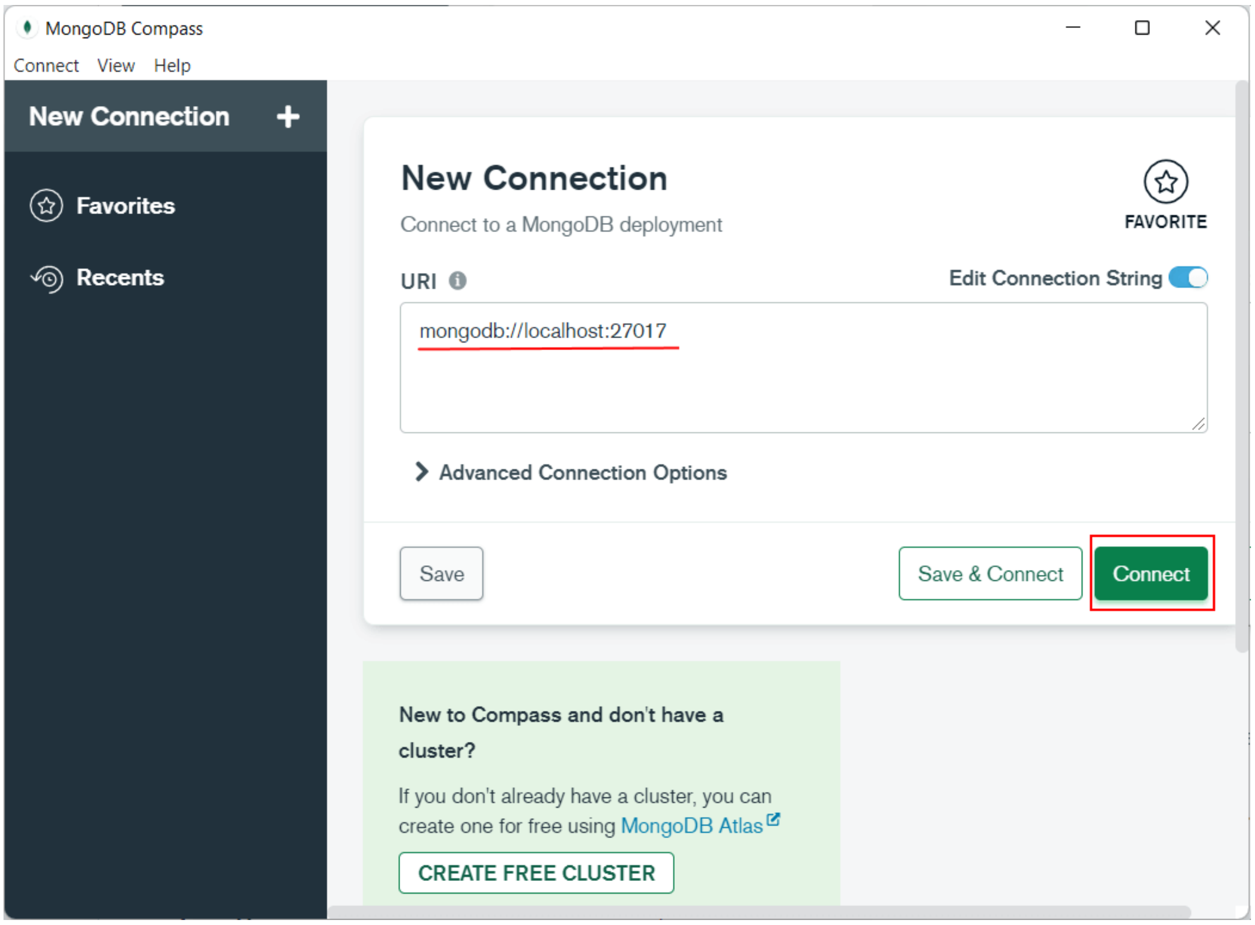
Version: 1.32.5 (Stable)

Platform: Windows 64-bit (7+) (Zip)

Package: zip

After downloading the zip archive, unzip it and go to the unzipped folder.

Here we can find the MongoDBCompass.exe file. It is he who represents the executable file of the program. Let's run it:



New Connection

Connect to a MongoDB deployment

URI

Edit Connection String

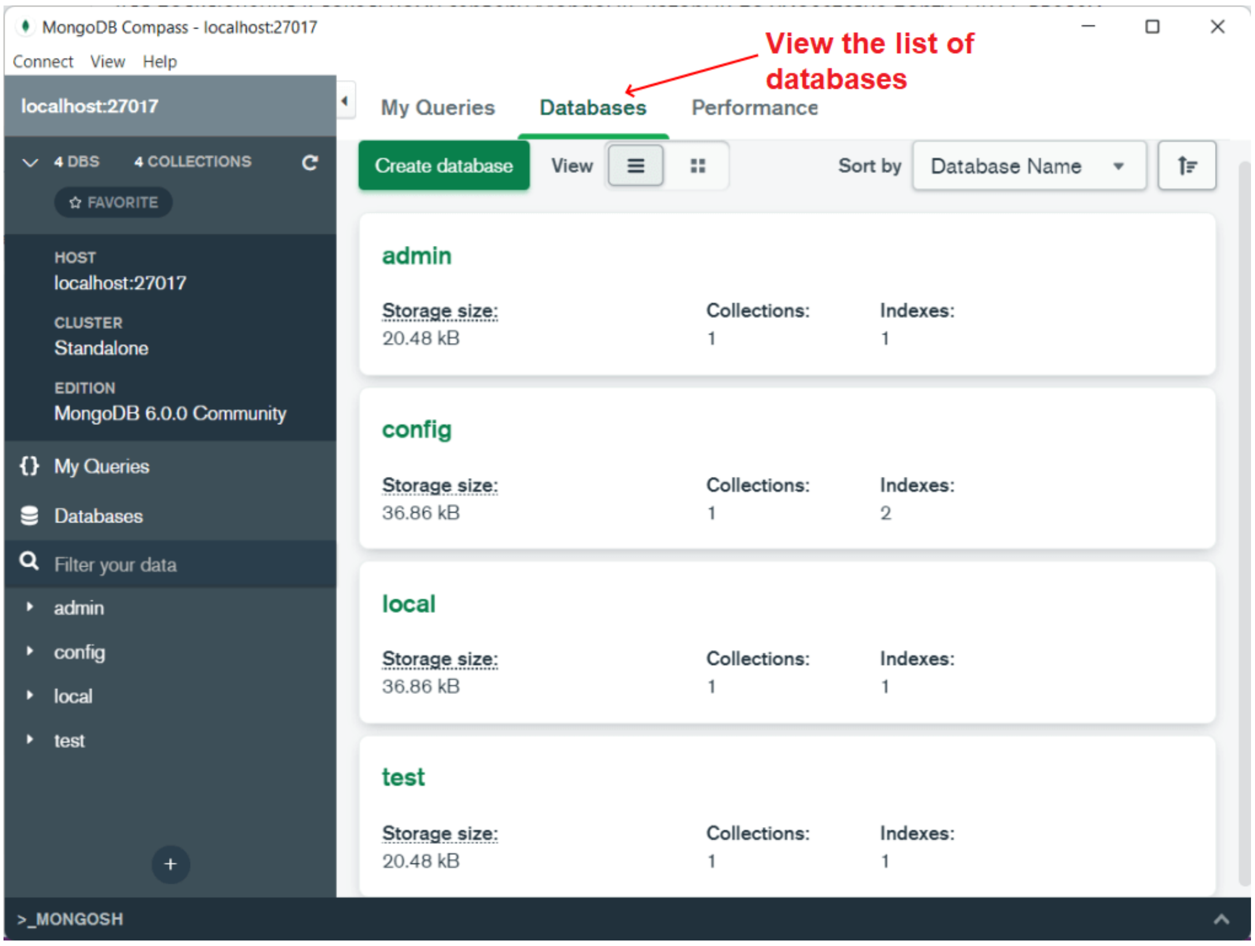
mongodb://localhost:27017

New to Compass and don't have a cluster?

If you don't already have a cluster, you can create one for free using MongoDB Atlas

By default, a window will open to create a new connection. It has only one field - URI, where you need to enter the connection string. By default, the connection line to the local server may appear here - mongodb://localhost:27017. The connection string may vary depending on where the MongoDB server is running (locally on the current computer or somewhere on another network resource), the login and password used, and other settings. In this case, we will consider connecting to a locally running mongodb server, the installation of which was discussed in the previous topic. Therefore, let's leave the value mongodb://localhost:27017 in it (or enter this line if the field is empty) and click the Connect button.

After a successful connection, the content of the server will open to us, where we can view various information, for example, the list of databases that are on the server:



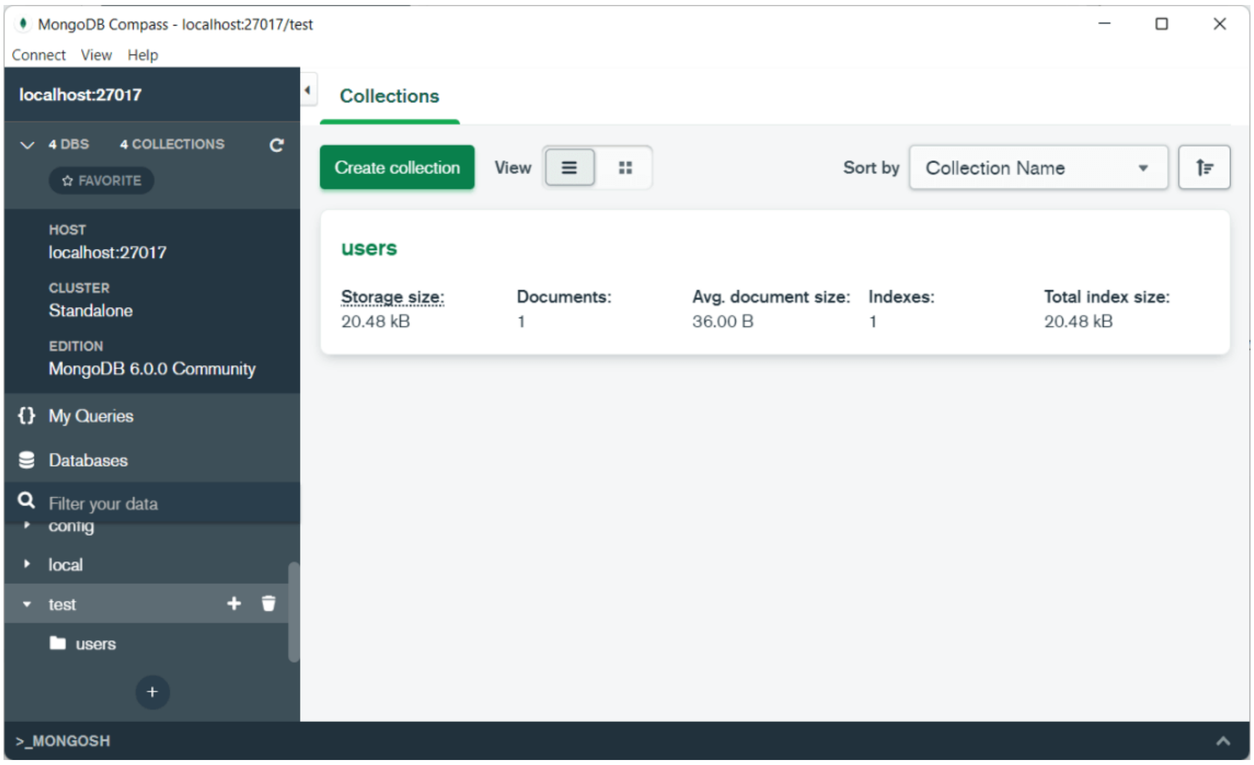
HOST
localhost:27017

CLUSTER
Standalone

EDITION
MongoDB 6.0.0 Community

	Storage size:	Collections:	Indexes:
admin	20.48 kB	1	1
config	36.86 kB	1	2
local	36.86 kB	1	1
test	20.48 kB	1	1

We can select a certain database and get information, in particular, see a set of collections in the database, how much data they occupy.


MongoDB Compass - localhost:27017/test

HOST
localhost:27017
CLUSTER
Standalone
EDITION
MongoDB 6.0.0 Community

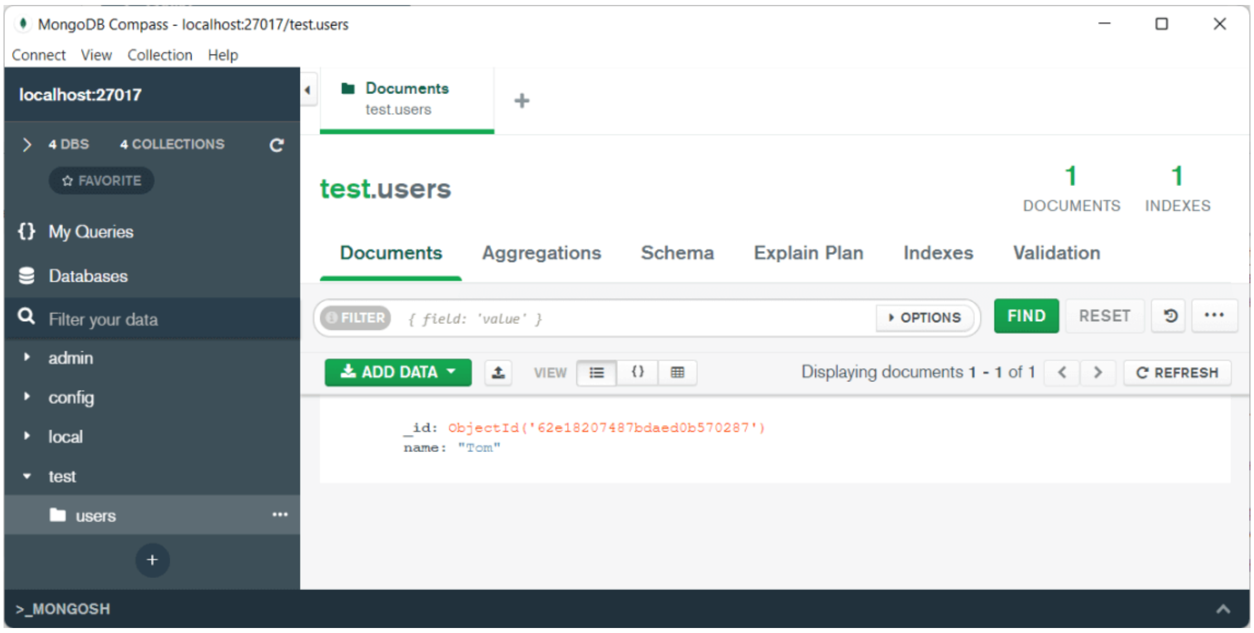
Sort by: Collection Name

users

Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20.48 kB	1	36.00 B	1	20.48 kB

In this case, the screenshot shows the list of test database collections created in the previous topic using the mongosh shell.

By clicking on a certain collection, you can graphically see all the data that is in the collection:


MongoDB Compass - localhost:27017/test.users

test.users

1 DOCUMENTS 1 INDEXES

Displaying documents 1 - 1 of 1

```
_id: ObjectId('62e18207487bdaed0b570287')
name: "Tom"
```

Using the Compass GUI, we can manage this data, add, change, delete it.