

В командной строке с помощью команды ниже создаем и запускаем контейнер с именем MONGO_CONTAINER в автономном режиме. Также скачивается MongoDB Docker image. Можно скачать этот образ заранее, воспользовавшись командой “ docker pull mongodb/mongodb-community-server:latest ”

```
C:\Users\User>docker run -d -p 27017:27017 --name MONGO_CONTAINER mongo:latest
Unable to find image 'mongo:latest' locally
latest: Pulling from library/mongo
bccd10f490ab: Already exists
b00c7ff578b0: Pull complete
a1f43ab85151: Pull complete
9e72f6a5998a: Pull complete
8424336879e4: Pull complete
85a6d3c2e6c8: Pull complete
c533c21e5fb8: Pull complete
1fddf702bb73: Pull complete
Digest: sha256:0e145625e78b94224d16222ff62609c4621ff6e2c390300e4e6bf698305596792
Status: Downloaded newer image for mongo:latest
4ed38bc6f7f480db1d1a2993ef7ca843d7ab7f25a1f54092dd66a99fe501b04e
```

Команда “docker ps” в командной строке позволяет увидеть все запущенные контейнеры. Мы видим, что запущен контейнер MONGO_CONTAINER, т.к. мы его запустили в прошлой команде.

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS | PORTS | NAME |
|----------------|---------------|--------------------------|-----------------|--------|-------|------|
| 15 minutes ago | Up 15 minutes | 0.0.0.0:27017->27017/tcp | MONGO_CONTAINER | | | |

Далее заходим в Docker в наш контейнер, переходим во вкладку “Ехес”, где прописываем “docker exec -it MONGO_CONTAINER bash”, нажимаем Enter и вводим “mongo”. С помощью этих действий мы можем работать в командной строке и через нее совершать всякие действия над контейнером.

Команда “show collections” показывает весь список коллекций (таблиц). С помощью команды “use dbTest” мы создаем базу данных с названием dbTest. В ней мы создаем таблицу с помощью “db.createCollection('myFirstCollection')”. С помощью команды “db.myFirstCollection.insert({item: "card1", qty: 10})” и “db.myFirstCollection.insert({item: "card2", qty: 20})” мы создаем столбцы item и qty, которым присваиваем свои значения, т.е. создаем строки. С помощью команды “db.myFirstCollection.find()” выводим содержимое коллекции (таблицы) myFirstCollection. Нам выводятся 2 ранее созданные строки. Причем заметим, что столбец id создается автоматически

```

test> show collections

test> use dbTest
switched to db dbTest
dbTest> db.createCollection('myFirstCollection')
{ ok: 1 }
dbTest> db.myFirstCollection.insert({item: "card1", qty: 10})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6603fd021ff09d9facdb83b0') }
}
dbTest> db.myFirstCollection.find()
[
  { _id: ObjectId('6603fd021ff09d9facdb83b0'), item: 'card1', qty: 10 }
]

dbTest> db.myFirstCollection.find()
[
  { _id: ObjectId('6603fd021ff09d9facdb83b0'), item: 'card1', qty: 10 },
  { _id: ObjectId('660407581ff09d9facdb83b1'), item: 'card2', qty: 20 }
]

```

Команда “show dbs” показывает список баз данных. Среди “классических” мы видим нашу dbTest.

```

dbTest> show dbs
admin    40.00 KiB
config  84.00 KiB
dbTest  72.00 KiB
local   40.00 KiB

```