

NoSQL

Module: 165

Utiliser des bases de données
NoSQL

Base de données NoSQL

Objectifs

2. Implémenter une base de données NoSQL et y insérer les données.
4. Sauvegarder la base de données NoSQL et vérifier sa restauration.

MongoDB


Sauvegarder et restaurer avec les outils MongoDB

MongoDB

Sauvegarde: *mongodump*

mongodump permettent de sauvegarder les bases de données du serveur MongoDB. Les base de données sont sauvegarder au format **BSON**.

mongodump peut soit sauvegarder toute une base de données, soit une ou des collections spécifié grâce à des options.




```
mongodump --uri=mongodb://root:admin@localhost:27017  
          --db="testdb"  
          --out=./backupdb/mongodump-testdb-20240121
```

MongoDB

Sauvegarde: *mongodump*

mongodump offre aussi la possibilité de transmettre les paramètres de connexion séparément, sans utiliser avec « --uri » la « connection string ».

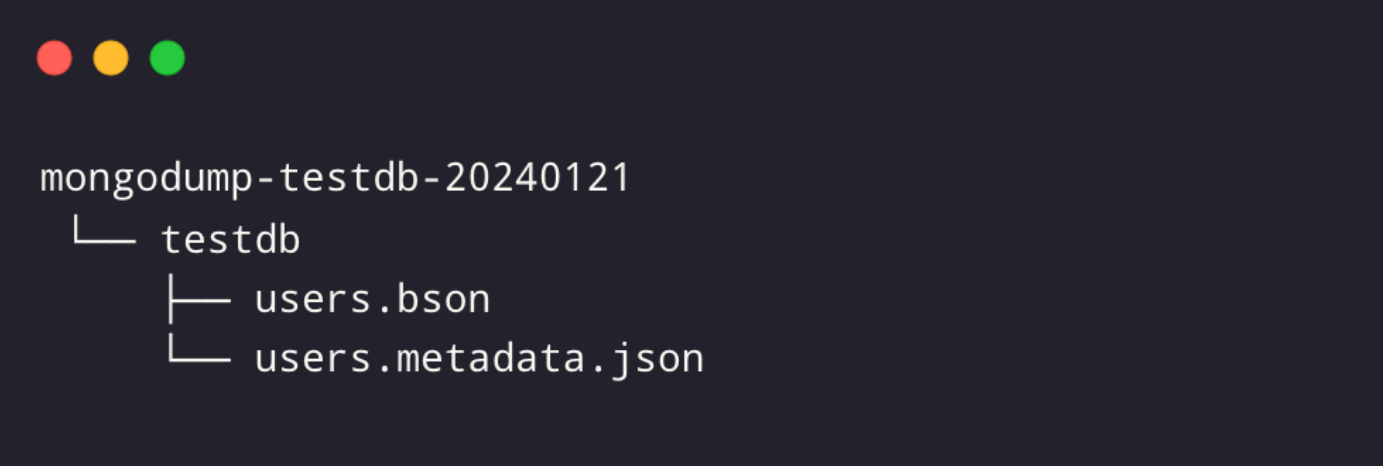


```
mongodump --host=localhost  
          --port=27017  
          --username=root  
          --password=<password>  
          --authenticationDatabase=admin  
          --db="testdb"  
          --out=./backupdb/mongodump-testdb-20240121
```

MongoDB

Sauvegarde: *mongodump*

Cela va créer un répertoire nommer 'testdb' dans le répertoire de destination « backupdb » lequel vous retrouverez 2 fichiers, correspondant à la collection 'users':



```
mongodump-testdb-20240121
├─ testdb
│   ├── users.bson
│   └─ users.metadata.json
```

- le fichier 'users.bson' contient les documents, donc les données au format **BSON**
- le fichier 'users.metadata.json', contient les définitions propres à la collection.

MongoDB

Sauvegarde: *mongodump*

mongodump permettent de sauvegarder les bases de données du serveur MongoDB. Les base de données sont sauvegarder au format **BSON**.

mongodump peut soit sauvegarder toute une base de données, soit une ou des collections spécifié grâce à des options.

Vous ne pouvez pas sauvegarder tout un serveur!



```
mongodump --uri=mongodb://root:admin@localhost:27017  
          --out=./backupdb/mongodump-20240121
```

MongoDB

Restorer: *mongorestore*

'mongorestore' permet de restaurer les bases de données avec les dumps créés avec 'mongodump' (fichier en **BSON**) de toutes les bases ou d'une base données spécifique ou d'une collection.



```
mongorestore --uri=mongodb://root:admin@localhost:27017 --authenticationDatabase admin dump
```

```
2023-10-17T13:25:55.395+0000 preparing collections to restore from
2023-10-17T13:25:55.395+0000 reading metadata for testdb.users from dump/testdb/users.metadata.json
2023-10-17T13:25:55.411+0000 restoring testdb.users from dump/testdb/users.bson
2023-10-17T13:25:55.426+0000 finished restoring testdb.users (2 documents, 0 failures)
2023-10-17T13:25:55.426+0000 no indexes to restore for collection testdb.users
2023-10-17T13:25:55.426+0000 2 document(s) restored successfully. 0 document(s) failed to restore.
```


MongoDB

Exporter: *mongoexport*

```
● ● ●  
  
mongoexport --uri=mongodb://root:admin@localhost:27017 --authenticationDatabase admin \  
            --db="testdb" --collection="users" -out="users.json"  
  
2023-10-17T13:35:58.625+0000    connected to: mongodb://[**REDACTED**]@localhost:27017  
2023-10-17T13:35:58.630+0000    exported 2 records  
  
cat users.json  
  
{ "_id": {"$oid": "652d566757e03b9b16eba164"}, "firstname": "Doe", "lastname": "John" }  
{ "_id": {"$oid": "652d5ed4f8b970048257fb03"}, "firstname": "Muller", "lastname": "Marie" }
```

Avec 'mongoexport' l'on doit spécifier la base de données et la collection que l'on désire exporter. Cela crée un fichier JSON uniquement avec les données / documents de la collection. Pas de meta-datas.

MongoDB

Exporter: *mongoexport*

```
mongoexport --uri=mongodb://root:admin@localhost:27017 --authenticationDatabase admin \
  -d testdb -c users \
  --type=csv --fields=firstname,lastname \
  -o users.csv

2023-10-17T13:48:57.817+0000    connected to: mongodb://[**REDACTED**]@localhost:27017
2023-10-17T13:48:57.823+0000    exported 2 records

cat users.csv

firstname,lastname
Doe,John
Muller,Marie
```

Il est aussi possible d'exporter dans le format CSV. Dans ce cas, il sera impératif de spécifier quelles 'fields' l'on désire exporter.

MongoDB

Importer: *mongoimport*

```
mongoimport --uri=mongodb://root:admin@localhost:27017 --authenticationDatabase admin \
--db="testdb" users.json

2023-10-17T13:43:00.834+0000 no collection specified
2023-10-17T13:43:00.834+0000 using filename 'users' as collection
2023-10-17T13:43:00.851+0000 connected to: mongodb://[**REDACTED**]@localhost:27017
2023-10-17T13:43:00.854+0000 2 document(s) imported successfully. 0 document(s) failed to import.
```

Avec 'mongoimport' l'on doit spécifier la base de données dans laquelle l'on désire importer la collection, en lui passant le fichier JSON. Si la base de données n'existe pas.

Attention! Les documents avec les mêmes **_id** ne seront pas mis à jour.

MongoDB

Sauvegarde et restauration: *résumé*

Au format **BSON**

mongodump
sauvegarde

mongorestore
restauration

Au format **JSON**

mongoexport
sauvegarde

mongoimport
restauration