# mongoDB

Une fois installé, il y a 2 fichiers : mongod.exe et mongo.exe

## mongod.exe

mongod.exe sert à démarrer le moteur de base de données.

Dans une invite de commande, pour lancer mongod, il suffit de taper mongod.exe.

Avant de lancer mongod.exe, il faut créer un répertoire pour stocker toutes les données.

Dans l’invite de commande : md \data\db

On peut utiliser un fichier de configuration, qui indiquera par exemple le dossier des données, le dossier des journaux (log), les autorisations etc.

|  |
| --- |
| *mongod.cfg – ancienne version toujours compatible*  logpath=C:\mongo\data\logs\mongo.log  dbpath=C:\mongo\data\db |

|  |
| --- |
| *mongod.cfg – version YAML*  storage:  dbPath: "\data\db"  directoryPerDB: true  journal:  enabled: true  systemLog:  destination: file  path: "\data\logs\mongo.log" |

Il faudra indiquer ce fichier lors du lancement de mongod.exe.

Dans l’invite de commande :

mongod.exe –config C:\mongodb\bin\mongod.cfg

Une fois le moteur lancé, un message indiquera le port d’écoute. Par défaut,  c’est le port 27017.

waiting for connections on port 27017

## mongo.exe

Une fois mongod.exe en cours d’exécution, il faut lancer une seconde invite de commande et lancer mongo.exe

db 🡪 refer to the current database

use student 🡪 connect to the database named ‘student’ 🡺 switched to db student

Collections ⬄ tables

Documents : objects in collections ⬄ records in a table

db.acteurs.insert({nom:"E.T.", terrien:false})

db : current database

acteurs : collection

{} : the document

db.acteur.find({}).count()

**Enable Auth**

Authentification : verify the identity of a user

Authorization : determine the verified user’s access to resources and operations

With access control enabled, ensure you have a user with userAdmin or userAdminAnyDatabase role in the admin database. This user can administrate user and roles such as: create users, grant or revoke roles from users, and create or modify customs roles.

To create userAdmin :

* Start MongoDB without access control (mongod.exe)

mongod --port 27017 --dbpath /data/db1

### Connect to the instance (mongo.exe)

### Create the user administrator

|  |
| --- |
| use admin  db.createUser(  {  user: "myUserAdmin",  pwd: "abc123",  roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]  }  ) |

🡺 La database où l’on crée le user est sa authentification database. On a crée le user dans une database spécifique, mais néanmoins, un user peut avoir des privilèges dans différentes databases en lui assignant des rôles dans d’autres databases. C’est le nom du user et sa authentification database qui est constitue l’identifiant unique du user. Ainsi, 2 users avec le même nom mais ayant été créée dans 2 databases différentes seront considérés comme 2 users différents.

Si un user doit avoir des droits sur plusieurs base de données, il faut le créer dans une database et ensuite lui donner les droits sur les autres databases plutôt que de créer le user dans chacune des databases.

### Disconnect the mongo shell

### Re-start the MongoDB instance with access control

mongod --auth --port 27017 --dbpath /data/db1

|  |
| --- |
| *mongod.cfg – version YAML*  storage:  dbPath: "\data\db"  directoryPerDB: true  journal:  enabled: true  systemLog:  destination: file  path: "\data\logs\mongo.log"  security:  authorization: "enabled" |

if a config file is used :

Clients that connect to this instance must now authenticate themselves as a MongoDB user

### Connect and authenticate as the user administrator

#### To authenticate during connection :

mongo --port 27017 -u "myUserAdmin" -p "abc123" --authenticationDatabase "admin"

#### To authenticate after connecting :

mongo --port 27017

use admin

db.auth("myUserAdmin", "abc123" )

### Create additional users as needed for your deployment.

use test

db.createUser(

{

user: "myTester",

pwd: "xyz123",

roles: [ { role: "readWrite", db: "test" },

{ role: "read", db: "reporting" } ]

}

)

### Connect and authenticate as myTester.