

Sécurité d'accès en MongoDB

Sécurité des accès

- Les fonctionnalités de contrôle d'accès permettent un accès par authentification aux utilisateurs existants,

Créer des utilisateurs

- Pour créer un utilisateur sur une base de données, il faut spécifier :
 - Le login
 - Le mot de passe
 - Et le (les) rôles à lui attribuer

Créer des utilisateurs

- Exemple :

```
use Examples
db.createUser(
  {user: "UserAdmin",
    pwd: "password" ,
    "roles" : [
      {
        "role" : "userAdmin",
        "db" : "Examples"
      }
    ]
  }
)
```

Les rôles d'un utilisateur

- Il existe des rôles prédéfinis :

Role	Explication
dbAdmin	Administrateur de la base de données
dbOwner	Combine les éléments suivants : readWrite, dbAdmin userAdmin
read	Les rôles disponibles au niveau de la base de données:
readWrite	<ul style="list-style-type: none">• read : Lire les données sur toutes les collections non-système• readWrite : en plus des privilèges du rôle "lecture", on a le droit d'écriture
userAdmin	créer et de modifier des users et des rôles sur la base de données actuelle

Créer des utilisateurs

- Pour vérifier la liste des utilisateurs créés au niveau de mongodb, on utilise:

```
db.system.users.find()
```

- Il faut se situer au niveau de la base system: **admin** pour exécuter la commande.

Afficher les détails d'un utilisateur

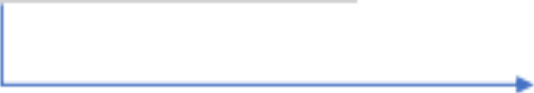
- Pour afficher les détails d'un user: `db.getUser(nomUtilisateur)`
- Exécuter la commande à partir de la base d'où on a créé l'utilisateur.

```
{
  "_id" : "Examples.UserAdmin",
  "userId" : UUID("8d6ea3c4-d054-447e-bde7-
cd5dc9de5140"),
  "user" : "UserAdmin",
  "db" : "Examples",
  "roles" : [
    {
      "role" : "userAdmin",
      "db" : " Examples "
    }
  ],
  "mechanisms" : [
    "SCRAM-SHA-1",
    "SCRAM-SHA-256"
  ]
}
```

Supprimer un utilisateur

- Pour supprimer un user: `db.dropUser(nomUtilisateur)`

```
db.dropUser("UserAdmin")  
db.getUser("UserAdmin")
```



```
switched to db Examples  
true  
nul
```


Création d'un rôle

- Pour créer un rôle, il faut préciser le nom du rôle et les privilèges de ce rôle.
- *Syntaxe:*

```
db.createRole ( { role : "NomDuRole" , privileges : [ {ressource : { db  
:"NomBase", collection : "NomCollection"} , actions : [actions1,action2,..] } ]  
, roles:[ ] } )
```

Les actions représentant les privilèges

- Action c'est l'opération ou la fonction que peut exécuter ce rôle. Il existe plusieurs , entre autres il y'a :
 1. **find**: l'utilisateur peut exécuter **aggregate()**, **find()**
 2. **insert**: l'utilisateur peut exécuter les commandes **d'insertion** et **create**
 3. **update**: l'utilisateur peut exécuter la commande **update**
 4. **remove** : l'utilisateur peut exécuter une suppression au niveau de la collection

Supprimer un rôle

- Pour supprimer un rôle personnalisé, on utilise :

```
db.dropRole(nomDuRole)
```