

Atelier I : Gestion d'une base de données MongoDB avec Studio 3T

Durée prévue : 2 heures

Objectif principal :

Apprendre à créer, gérer et manipuler une base de données MongoDB en utilisant Studio 3T, à travers un exemple basé sur un système de gestion des utilisateurs.

1. Introduction à MongoDB et Studio 3T (15 minutes)

1. Qu'est-ce que MongoDB ?

- MongoDB est une base de données orientée documents NoSQL, idéale pour gérer des données semi-structurées ou volumineuses.
- **Caractéristiques clés :**
 - Les documents (JSON-like) permettent une flexibilité de schéma.
 - Adaptée pour les applications nécessitant une scalabilité horizontale.

2. Pourquoi Studio 3T ?

- Studio 3T est un outil graphique avancé pour MongoDB :
 - Simplifiez la gestion des bases de données.
 - Intégrer IntelliShell pour exécuter des commandes.
 - Outils visuels pour explorer et manipuler les données.

2. Cas pratique : Gestion des utilisateurs d'une plateforme en ligne

Une plateforme en ligne doit gérer les données de ses utilisateurs. Chaque utilisateur a des informations telles que son nom, son email, son rôle (ex. : "admin" ou "user"), et ses préférences.

3. Prise en main de Studio 3T et configuration initiale (20 minutes)

1. Connexion à MongoDB :

- Dans Studio 3T :
 - Configurer une nouvelle connexion.
 - Vérifiez la connexion avec l'instance MongoDB.

2. Créer une base de données :

- Ouvrez IntelliShell en cliquant avec le bouton droit sur la connexion principale et sélectionnez **"Ouvrir IntelliShell"** .
- **Commande :**

```
use UserManagement;
```

- **Explication :**

- Cette commande crée ou sélectionne la base de données UserManagement.

3. Créer une collection pour les utilisateurs :

- **Commande :**

```
db.createCollection("Users");
```



- Crée une collection appelée `Users`.

4. Insérer des données utilisateurs (25 minutes)

1. Insertion de plusieurs utilisateurs :

- **Scénario :**

```
db.Users.insertMany([
  { name: "Karim Alami", email: "karim@example.com", role:
    "admin", preferences: { theme: "dark", notifications: true } },
  { name: "Said Taha", email: "said@example.com", role: "user",
    preferences: { theme: "light", notifications: false } },
  { name: "Fatima Farah", email: "fatima@example.com", role:
    "user", preferences: { theme: "dark", notifications: true } }
]);
```

- **Explication :**

- Chaque document représente un utilisateur.
- Les champs incluent des données imbriquées pour les préférences.

2. Exécution du script :

- Cliquez sur **"Run Entire Script"** pour insérer les documents.

3. Rafraîchir et vérifier les données :

- Rafraîchir l'arborescence pour afficher la collection `Users` dans Studio 3T.
- Explorer les documents insérés dans l'interface graphique.

5. Manipuler et analyser les données (40 minutes)

1. Lire les données :

- **Commande :**

```
db.Users.find();
```

- Affiche tous les documents de la collection.

2. Filtrer les utilisateurs :

- Trouver les utilisateurs avec le thème "dark" :

Copier le code

```
db.Users.find({ "preferences.theme": "dark" });
```

- **Explication :**

- La notation `"preferences.theme"` cible les champs imbriqués.

3. Mise à jour d'un utilisateur :

- Mettre à jour le rôle de Bob en "modérateur" :

```
db.Users.updateOne(
  { email: "said@example.com" },
  { $set: { role: "moderator" } }
```



);

- **Explication :**

- Le filtre `email` identifie l'utilisateur.
- `$set` met à jour ou ajoute un champ.

4. Suppression d'un utilisateur :

- Supprimer l'utilisateur "Charlie Brown" :

```
db.Users.deleteOne({ name: " Fatima Farah" });
```

- Supprimez le document correspondant.

5. Ajout d'un champ à tous les utilisateurs :

- Ajouter un champ `status` avec la valeur "active" :

```
db.Users.updateMany({}, { $set: { status: "active" } });
```

- **Explication :**

- La condition `{}` cible tous les documents.

6. Création d'un rapport avec l'agrégation (20 minutes)

1. Compte des utilisateurs par rôle :

- **Commande :**

```
db.Users.aggregate([  
  { $group: { _id: "$role", count: { $sum: 1 } } }  
]);
```

- **Explication :**

- `$group` regrouper les documents par `role`.
- `$sum` calcule le nombre d'utilisateurs par rôle.

2. Exporter les données :

- Utilisez les fonctionnalités de Studio 3T pour exporter les résultats de la collection ou de l'agrégation dans un fichier CSV.

7. Conclusion et questions (15 minutes)

- **Résumé des apprentissages :**

- Création d'une base de données MongoDB.
- Gestion des collections et insertion de données significatives.
- Manipulation, mise à jour et suppression des documents.
- Génération de rapports simples avec les pipelines d'agrégation.

- **Discussion :**

- Importance de MongoDB dans les applications modernes.
- Cas d'utilisation des bases de données NoSQL.
- Questions-réponses des participants.