

TP PRÉPARATOIRE : Initialisation à MongoDB Atlas

Contexte

Avant d'aborder la communication complexe entre votre API et la base de données demain, nous allons simplement préparer le terrain. Ce TP a pour unique but de valider que votre compte cloud est actif et que votre ordinateur a le droit d'y accéder.

Durée estimée : 30 minutes

Objectif : Obtenir une connexion "Succès" dans le terminal.

ÉTAPE 1 : Configuration du Cloud (MongoDB Atlas)

Nous n'installerons rien de lourd sur votre machine. Nous utiliserons le service cloud officiel.

1. Crédit de compte

Rendez-vous sur le site officiel de MongoDB Atlas et créez un compte gratuit.

2. Crédit de la base de données (Cluster)

- Sélectionnez l'offre "Shared" (Gratuite / Free Tier).
- Fournisseur : AWS (par défaut).
- Région : Choisissez Paris (eu-west-3) si disponible, ou Francfort.
- Nom du Cluster : Laissez "Cluster0" ou nommez-le "MernStarter".
- Validez la création (cela prend 2 à 3 minutes).

3. Sécurité et Accès (CRUCIAL)

Pendant la création du cluster, une étape de sécurité s'affiche :

- **Utilisateur** : Créez un utilisateur (ex: admin).
- **Mot de passe** : Générez un mot de passe et **copiez-le immédiatement** dans un bloc-notes. Vous ne pourrez plus le voir ensuite.
- **IP Access** : Ajoutez votre adresse IP actuelle (bouton "Add my current IP Address").
Pour éviter tout blocage si vous changez de wifi, vous pouvez aussi ajouter 0.0.0.0/0 (autorise tout le monde).

4. Récupération du lien de connexion

- Une fois le cluster vert (actif), cliquez sur le bouton "Connect".
- Choisissez "Drivers".
- Copiez la chaîne de connexion sous "Add your connection string into your application code".
- Elle ressemble à :
`mongodb+srv://admin:<password>@cluster0.p8plk.mongodb.net/?retryWrites=true&w=majority`

ÉTAPE 2 : Préparation du projet de test

Créez un dossier nommé test-mongo sur votre bureau et ouvrez-le avec VS Code.

Dans le terminal intégré de VS Code, lancez ces commandes une par une :

1. Initialisation du projet :

```
npm init -y
```

2. Installation des outils TypeScript et MongoDB (Mongoose) :

```
npm install mongoose
```

```
npm install typescript ts-node @types/node @types/mongoose --save-dev
```

3. Initialisation de la configuration TypeScript :

```
npx tsc --init
```

ÉTAPE 3 : Le Script de Connexion

Créez un fichier nommé start.ts à la racine du projet.

Copiez le code ci-dessous. Il est volontairement très simple pour éviter les erreurs de typage complexes pour ce soir.

```
import mongoose from 'mongoose';

// REMPLACEZ LA LIGNE CI-DESSOUS PAR VOTRE PROPRE LIEN ATLAS
// N'oubliez pas de remplacer <password> par votre vrai mot de passe
const uri =
"mongodb+srv://admin:votreMotDePasse@cluster0.xxxxx.mongodb.net/?retryWrites=true&w=majority";

async function testConnection() {
  try {
    console.log("Tentative de connexion...");
    // 1. Connexion
    await mongoose.connect(uri);
    console.log("Connexion réussie à MongoDB Atlas !");
    // 2. Définition simple d'un format de données (Schema)
    const CatSchema = new mongoose.Schema({
      name: String
    });
  }
}
```

```

// 3. Création du modèle
const Cat = mongoose.model('Cat', CatSchema);

// 4. Création et sauvegarde d'une donnée simple
const myCat = new Cat({ name: 'Zizou' });
await myCat.save();

console.log("Donnée sauvegardée : Un chat nommé Zizou a été ajouté.");

} catch (error) {
  console.error("Erreur de connexion :", error);
} finally {
  // Fermeture propre
  await mongoose.disconnect();
  console.log("Fermeture de la connexion.");
}
}

testConnection();

```

ÉTAPE 4 : Validation

1. Dans le fichier start.ts, assurez-vous d'avoir collé votre lien entre les guillemets de la variable uri.
2. Remplacez bien <password> par le mot de passe créé à l'étape 1 (sans les chevrons <>).
3. Lancez le script via le terminal :

npx ts-node start.ts

Résultat attendu :

Si le terminal affiche "Connexion réussie..." puis "Donnée sauvegardée...", c'est gagné.

Si vous avez une erreur bad auth, c'est souvent le mot de passe qui est incorrect ou des caractères spéciaux dans le mot de passe qui gênent l'URL.

Si vous avez une erreur de timeout, c'est souvent l'adresse IP qui n'a pas été autorisée (Step 1.3).

Une fois que cela fonctionne, vous êtes prêts pour le cours de demain.