

# Cours complet sur Prisma

KONDI Abdoul Malik

IFNTI

November 7, 2024

## Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Introduction à Prisma</b>                                | <b>2</b> |
| 1.1      | Qu'est-ce que Prisma ?                                      | 2        |
| 1.2      | Les avantages de Prisma                                     | 2        |
| <b>2</b> | <b>Installation de Prisma</b>                               | <b>2</b> |
| 2.1      | Prérequis   | 2        |
| 2.2      | Installation  | 2        |
| <b>3</b> | <b>Le Schéma Prisma</b>                                     | <b>2</b> |
| 3.1      | Définition des modèles                                      | 2        |
| 3.2      | Types et Relations  | 2        |
| <b>4</b> | <b>Exécuter des Migrations</b>                              | <b>3</b> |
| 4.1      | Création d'une migration                                    | 3        |
| <b>5</b> | <b>Prisma Client</b>  | <b>3</b> |
| 5.1      | Générer Prisma Client                                       | 3        |
| 5.2      | Utilisation de Prisma Client                                | 3        |
| <b>6</b> | <b>Les Seeders dans Prisma</b>                              | <b>3</b> |
| 6.1      | Qu'est-ce qu'un Seeder ?                                    | 3        |
| 6.2      | Créer un Seeder avec Prisma                                 | 3        |
| <b>7</b> | <b>Génération de la Documentation de la Base de Données</b> | <b>4</b> |
| 7.1      | Prisma Studio   | 4        |
| 7.2      | Générer la Documentation avec Prisma ERD Generator          | 4        |
| 7.3      | Installation de l'outil ERD                                 | 4        |
| <b>8</b> | <b>Travaux Pratiques</b>                                    | <b>4</b> |
| 8.1      | TP1 : Modélisation d'une base de données                    | 4        |
| 8.2      | TP2 : Implémentation des requêtes avec Prisma Client        | 4        |
| 8.3      | TP3 : Génération de la documentation                        | 4        |
| <b>9</b> | <b>Conclusion</b>   | <b>4</b> |

# 1 Introduction à Prisma

## 1.1 Qu'est-ce que Prisma ?

Prisma est un ORM (Object-Relational Mapping) moderne pour Node.js et TypeScript. Il permet une interaction fluide avec les bases de données relationnelles comme PostgreSQL, MySQL, et SQLite.

## 1.2 Les avantages de Prisma

- Type safety : Prisma génère des types TypeScript pour votre base de données.
- Migrations simplifiées : Prisma gère les migrations de schéma.
- Intégration avec les bases de données : Prisma prend en charge les bases de données populaires.
- Requêtes optimisées : Prisma simplifie l'écriture des requêtes complexes.

# 2 Installation de Prisma

## 2.1 Prérequis

- Node.js installé ( $\geq 12.6.0$ )
- Un gestionnaire de paquets comme npm ou Yarn

## 2.2 Installation

```
npm install prisma --save-dev
npx prisma init
```

Cela crée un fichier `prisma/schema.prisma`, où vous définissez le modèle de votre base de données.

# 3 Le Schéma Prisma

## 3.1 Définition des modèles

Les modèles Prisma définissent les tables et les colonnes dans votre base de données.

```
model User {
  id      Int      @id @default(autoincrement())
  email   String   @unique
  name    String?
  posts   Post[]
}

model Post {
  id      Int      @id @default(autoincrement())
  title   String
  content String?
  author  User      @relation(fields: [authorId], references: [id])
  authorId Int
}
```

## 3.2 Types et Relations

- **Types de base** : Int, String, Boolean, DateTime.
- **Relations** : Un utilisateur peut avoir plusieurs posts.

## 4 Exécuter des Migrations

### 4.1 Création d'une migration

```
npx prisma migrate dev --name init
```

Cela crée une migration initiale.

## 5 Prisma Client

### 5.1 Générer Prisma Client

```
npx prisma generate
```

Cela génère un client pour interagir avec la base de données.

### 5.2 Utilisation de Prisma Client

```
import { PrismaClient } from '@prisma/client'
const prisma = new PrismaClient()

const allUsers = await prisma.user.findMany()
```

## 6 Les Seeders dans Prisma

### 6.1 Qu'est-ce qu'un Seeder ?

Un seeder permet de remplir votre base de données avec des données initiales ou fictives. Prisma ne fournit pas de fonctionnalité de seeding par défaut, mais vous pouvez utiliser un script pour en créer.

### 6.2 Créer un Seeder avec Prisma

Voici un exemple pour insérer des données initiales dans la base de données :

```
import { PrismaClient } from '@prisma/client'
const prisma = new PrismaClient()

async function main() {
  await prisma.user.createMany({
    data: [
      { email: 'alice@example.com', name: 'Alice' },
      { email: 'bob@example.com', name: 'Bob' }
    ],
  })

  await prisma.post.create({
    data: {
      title: 'Premier post',
      content: 'Ceci est un post',
      author: { connect: { email: 'alice@example.com' } }
    }
  })
}

main()
  .catch((e) => console.error(e))
  .finally(async () => {
    await prisma.$disconnect()
  })
```

Ensuite, exécutez le script avec :

```
npx ts-node prisma/seed.ts
```

## 7 Génération de la Documentation de la Base de Données

### 7.1 Prisma Studio

Prisma Studio est un outil visuel pour interagir avec votre base de données. Vous pouvez y explorer vos données directement depuis l'interface.

### 7.2 Générer la Documentation avec Prisma ERD Generator

Vous pouvez générer une documentation visuelle de votre base de données avec un outil comme `prisma-erd-generator`.

### 7.3 Installation de l'outil ERD

```
npm install prisma-erd-generator
```

Ensuite, dans votre fichier `schema.prisma`, ajoutez cette configuration pour générer l'ERD (Diagramme Entité-Relation) :

```
generator erd {  
  provider = "prisma-erd-generator"  
  output   = "../prisma/ERD"  
}
```

Pour générer le diagramme ERD :

```
npx prisma generate
```

Cela créera un fichier dans le répertoire spécifié qui contient le diagramme de vos modèles.

## 8 Travaux Pratiques

### 8.1 TP1 : Modélisation d'une base de données

1. Créez un modèle `Product` avec les champs suivants : `id`, `name`, `price`, et `stock`.
2. Mettez en place une relation avec un modèle `Category`.

### 8.2 TP2 : Implémentation des requêtes avec Prisma Client

1. Écrivez des requêtes pour insérer des produits dans la base de données.
2. Créez une requête pour récupérer tous les produits avec un prix supérieur à 50.

### 8.3 TP3 : Génération de la documentation

1. Utilisez `prisma-erd-generator` pour générer un diagramme de votre base de données.
2. Documentez votre base de données et présentez un rapport avec les relations principales.

## 9 Conclusion

Prisma est un outil puissant pour la gestion des bases de données dans des applications Node.js, avec une intégration fluide avec TypeScript, des migrations automatisées, et des fonctionnalités avancées de requêtes.