

1. Configuração Inicial

Primeiro, certifique-se de que você já tem o Prisma instalado e configurado conforme mencionado anteriormente.

2. Definição do Esquema

Vamos criar um esquema básico para um sistema de blog com usuários e posts. Crie um arquivo `schema.prisma` no diretório `prisma/`:

```
prisma

// prisma/schema.prisma

datasource db {
  provider = "mysql"
  url      = env("DATABASE_URL")
}

generator client {
  provider = "prisma-client-js"
}

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

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

3. Migração do Banco de Dados

Para aplicar as mudanças no banco de dados, execute as migrações:

```
sh
```

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

4. Uso do Prisma Client

Agora, você pode usar o Prisma Client para interagir com o banco de dados. Aqui está um exemplo de como criar um novo usuário e postagem:

```
javascript
```

```
// src/app.js
```

```
const { PrismaClient } = require('@prisma/client');
```

```
const prisma = new PrismaClient();
```

```
app.post('/users', async (req, res) => {
```

```
  try {
```

```
    const user = await prisma.user.create({
```

```
      data: req.body,
```

```
    });
```

```
    res.status(201).json(user);
```

```
  } catch (error) {
```

```
    res.status(500).json({ error: 'Failed to create user' });
```

```
  }
```

```
});
```

```
app.post('/posts', async (req, res) => {
```

```
  try {
```

```
    const post = await prisma.post.create({
```

```
      data: {
```

```
        title: req.body.title,
```

```
        content: req.body.content,
```

```
        author: {
```

```
          connect: {
```

```
            id: req.body.authorId,
```

```

    },
  },
});
res.status(201).json(post);
} catch (error) {
  res.status(500).json({ error: 'Failed to create post' });
}
});

```

5. Consulta de Dados

Aqui está um exemplo de como consultar usuários e postagens:

javascript

```

app.get('/users', async (req, res) => {
  try {
    const users = await prisma.user.findMany();
    res.json(users);
  } catch (error) {
    res.status(500).json({ error: 'Failed to fetch users' });
  }
});

```

```

app.get('/posts', async (req, res) => {
  try {
    const posts = await prisma.post.findMany();
    res.json(posts);
  } catch (error) {
    res.status(500).json({ error: 'Failed to fetch posts' });
  }
});

```

6. Configuração do .env

Certifique-se de que seu arquivo .env esteja configurado corretamente:

DATABASE_URL="mysql://USER:PASSWORD@HOST:PORT/DATABASE"

7. Execução do Projeto

Finalmente, execute seu projeto:

```
sh
```

```
node server.js
```