

Into to Building an API With



Prisma

Hello!

I am Dom!



Proud to serve as an Instructor Manager at Lighthouse Labs, where I hope to inspire and mentor the next generation of tech talent.



dominic@lighthouselabs.com



[/in/dominictremblay/](https://www.linkedin.com/in/dominictremblay/)

What is Prisma

A modern Node.js and TypeScript
ORM (Object Relational Mapper)



Benefits of Prisma



Easy to Write Queries

Prisma **reduces the effort** needed to write complex queries through a powerful query builder.



Typescript

Prisma's type-safe API generates TypeScript code for database models, and help to **reduce errors** and **improve the overall quality** of code.



Enhanced security

Prisma automatically generates **secure SQL queries** that protect against SQL injection attacks.



Human-Readable Schema

The Prisma schema is **intuitive** and lets you declare your database tables in a **human-readable** way



Visual database browser

Prisma Studio is the **easiest way to explore** and manipulate data in your Prisma projects



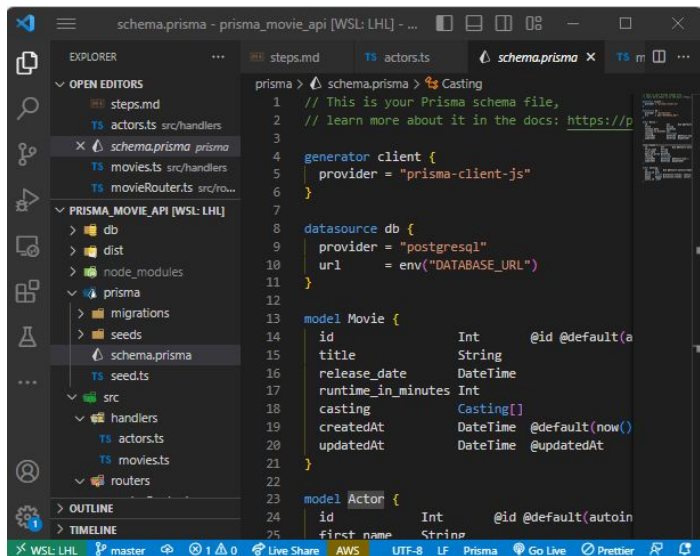
Scalability

Work **seamlessly with modern databases**, such as MySQL, PostgreSQL, and MongoDB, making it easy to **scale** applications as needed.

Demo Objectives

- ① Understand the basics of Prisma as an ORM tool
- ① Learn how to set up Prisma in a Node.js project (asynchronously)
- ① Use Prisma to perform basic CRUD (Create, Read, Update, Delete) operations

Time for Demo!



The screenshot shows a VS Code editor window with the following content:

- EXPLORER:** Shows the project structure for `schema.prisma - prisma_movie_api [WSL: LHL]`. The `prisma` folder is expanded, showing `migrations`, `seeds`, and `schema.prisma`.
- EDITOR:** Displays the `schema.prisma` file with the following content:

```
1 // This is your Prisma schema file,
2 // learn more about it in the docs: https://prisma.io/docs
3
4 generator client {
5   provider = "prisma-client-js"
6 }
7
8 datasource db {
9   provider = "postgresql"
10  url       = env("DATABASE_URL")
11 }
12
13 model Movie {
14   id           Int           @id @default(autoincrement())
15   title        String
16   release_date DateTime
17   runtime_in_minutes Int
18   casting      Casting[]
19   createdAt    DateTime @default(now())
20   updatedAt    DateTime @updatedAt
21 }
22
23 model Actor {
24   id           Int           @id @default(autoincrement())
25   first_name   String
```
- OUTPUT:** Shows the generated Prisma client code for the `Movie` model.

Key Takeaways

- ◎ Very easy to create data models
- ◎ Prisma makes it easy to write queries
- ◎ TypeScript will catch errors upfront
- ◎ Migration System
- ◎ Great visual browser



Thanks!

Any questions?