

Saved

-  **Read later**
-  Archive
-  Annotations
-  Web pages
-  Uploads
-  Intelligence reports

Tags

-  All tagged articles
-  Qwenta

Read later

+ Add

Archive all

Continue reading

All articles

 Search in articles



	SOURCE	LENGTH	PROGRESS	DATE SAVED ▾
	Update terminology in REST Security Cheat Sheet	OWASP Cheat Shee... N/A	0 %	Sun Sep 06, 2026 17:26
	Building a Modern Backend with TypeScript, PostgreSQL and Prisma: REST API, Validation, and Testing	Prisma Blog 01:00	0 %	Sun Sep 06, 2026 17:06

No more articles here.










[← Back to blog](#)

Backend with TypeScript, PostgreSQL & Prisma: REST, Validation & Tests

**Daniel Norman**

August 20, 2020

Updated July 3, 2026

Part 2 of 4 in the [Building a modern backend with TypeScript, PostgreSQL and Prisma](#) series. [View full series →](#)

Update (July 2026): This article was updated for Prisma ORM 7. The REST API is now built with [Hono](#) instead of Hapi, with [Zod](#) for input validation and [Vitest](#) for testing. It uses the rust-free `prisma-client` generator, the `prisma.config.ts` config file, and driver adapters. All commands and code work verbatim on Prisma ORM 7 with Node.js 20 or later.

This article is part of a series on building a backend with TypeScript, PostgreSQL, and Prisma. In this second part, you will build a REST API on top of the data model from part 1, validate input, and test the API.

Introduction

On this page

Introduction

[What the series covers](#)[What you will learn today](#)

Prerequisites

[Assumed knowledge](#)[Development environment](#)

Building a REST API

[The API endpoints](#)[Hono](#)[Making Prisma Client available with middleware](#)[Creating the server](#)[Defining the status route](#)[Testing the status route](#)[Defining the user routes](#)