Engineer Data

Designed a JSON passthrough from Postgres to API

Designed a JSON passthrough architecture where PostgreSQL functions return complete JSON forwarded verbatim by the Go API, eliminating intermediate unmarshalling and decoupling the frontend from schema changes.

2025

JSON straight from the database to the browser

Situation. The usual way data gets from a database to a browser is a relay race of transformations. The database hands the API rows, the API unmarshals them into structs, reshapes them, serialises them back to JSON, and only then do they go out. Every one of those hops is code you write, code you test, and one more place where the API's idea of the data and the database's idea of it can drift apart.

Task. The idea was to skip the relay race. If the database could return the finished response, the API could just pass it along, and the frontend could depend on the database's shape directly instead of on a hand-maintained copy of it living in Go.

Action. So it was built as a straight passthrough. The PostgreSQL functions assemble the whole response as JSON — the shaping is a SQL concern, done where the data already is. The Go handler takes that back as json.RawMessage and forwards it untouched; it never decomposes it, never re-encodes it. A small QueryJSON helper made that pattern the path of least resistance rather than something you had to remember to do. What fell out was all the intermediate machinery a conventional layered API collects — the DTOs, the mappers, the response structs.

Result. Handler code got dramatically shorter, and more to the point the frontend stopped being coupled to Go. Change what a function returns and the new shape flows straight through to the client without anyone editing a line

of handler code. Fewer moving parts, and one whole category of layer-to-layer drift simply doesn't exist here.

Bring this experience to your project

Use "Designed a JSON passthrough from Postgres to API" as a starting point for a similar challenge in your organisation. The services behind this work are available for your project. A structure that matches how the business actually works, so the questions you need answered later are still possible to ask.

Explore our services Get in touch →

Look behind the result

Read the situation, decisions and result in this example. We can discuss which parts transfer to your constraints and which need another approach. Designed a JSON passthrough from Postgres to API — JSON straight from the database to the browser

Explore our services Get in touch →

Want your API and database in effortless sync?

Talk through your project with the engineer responsible for delivery. We agree the scope, price basis and handover in writing before work begins.

Role context: Platform Architect. It draws on Platform Architecture, Backend Engineering, APIs & Integration, Databases, PostgreSQL, SQL and Performance Tuning. And while you are here, we are rather good at Platform & Solution Architecture, Backend & API Development, AI Agent Tools & MCP Servers and Database Design & Modeling.

Get in touch Explore our services

One piece of **a longer record** — and it is all on this site.

Every entry is written the same way: the situation, the task, what we did and what changed.

See the whole record

Define the improvement you need

Bring a representative dataset, the questions it must answer and the current bottleneck. We can agree how to measure quality, throughput and recovery.

See what it costs Get in touch →

Want your API and database in effortless sync?

Ask about the decisions behind “Designed a JSON passthrough from Postgres to API”. You will speak with the people who can explain the work and assess your own project.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://data.engineer.company/portfolio/designed-a-json-passthrough-from-postgres-to-api/