Engineer Data

Har designet en JSON-passthrough-arkitektur

Har designet en JSON passthrough-arkitektur, hvor PostgreSQL-funktioner returnerer komplet JSON, som Go-API'et videresender uændret, så der ikke er nogen mellemliggende unmarshalling, og frontenden er afkoblet fra skemaændringer.

2025

JSON direkte fra databasen til browseren

Situation. Normalt kommer data fra en database til en browser gennem et stafetløb af transformationer. Databasen giver API'et rækker, API'et unmarshaller dem til structs, omformer dem, serialiserer dem tilbage til JSON, og først derefter bliver de sendt ud. Hvert af de led er kode, man skriver, kode, man tester, og endnu et sted, hvor API'ets opfattelse af dataene og databasens kan glide fra hinanden.

Opgave. Idéen var at springe stafetløbet over. Hvis databasen kunne returnere det færdige svar, kunne API'et bare sende det videre, og frontenden kunne bygge direkte på databasens struktur i stedet for på en håndvedligeholdt kopi af den i Go.

Handling. Så det blev bygget som en ren passthrough. PostgreSQL-funktionerne samler hele svaret som JSON — formgivningen er en SQL-opgave og sker der, hvor dataene allerede ligger. Go-handleren tager imod det som json.RawMessage og sender det videre uændret; den pakker det aldrig ud og koder det aldrig om. En lille QueryJSON-helper gjorde mønstret til den letteste vej i stedet for noget, man skulle huske at gøre. Det, der faldt bort, var alt det mellemliggende maskineri, som et konventionelt lagdelt API samler på sig — DTO'erne, mapperne, svar-structene.

Resultat. Handler-koden blev markant kortere, og hvad der vejer tungere: frontenden holdt op med at være koblet til Go. Ændrer man, hvad en funktion returnerer, løber den nye struktur direkte igennem til klienten, uden at

nogen retter en linje handler-kode. Færre bevægelige dele, og en hel kategori af afvigelser mellem lagene findes simpelthen ikke her.

Brug erfaringen i jeres projekt

Brug »Har designet en JSON-passthrough-arkitektur« som udgangspunkt for en lignende udfordring i jeres organisation. Ydelserne bag arbejdet kan også bruges i jeres projekt. En struktur, der passer til, hvordan forretningen faktisk fungerer, så de svar, I får brug for senere, stadig kan hentes.

Se vores ydelser Kontakt os →

Se bag resultatet

Læs om situationen, beslutningerne og resultatet i eksemplet. Vi kan drøfte, hvad der passer til jeres vilkår, og hvad der kræver en anden tilgang. Har designet en JSON-passthrough-arkitektur — JSON direkte fra databasen til browseren

Se vores ydelser Kontakt os →

Vil du have API og database til at følges ad uden besvær?

Tal om jeres projekt med den ingeniør, der har ansvaret for leverancen. Vi aftaler omfang, prisgrundlag og overdragelse skriftligt, før arbejdet begynder.

Rolle i opgaven: Platformarkitekt. Det trækker på Platformarkitektur, Backend-udvikling, API'er & integration, Databaser, PostgreSQL, SQL og Optimering af ydeevne. Og når I alligevel er her, er vi ret dygtige til Platform- & løsningsarkitektur, Backend- & API-udvikling, Værktøjer til AI-agenter & MCP-servere og Databasedesign & datamodellering.

Kontakt os Se vores ydelser

Ét stykke af **en længere fortegnelse** — og det hele ligger her på sitet.

Hver post er skrevet på samme måde: situationen, opgaven, hvad vi gjorde, og hvad der ændrede sig.

Se hele fortegnelsen

Definér den ønskede forbedring

Medbring et repræsentativt datasæt, de spørgsmål det skal besvare, og den aktuelle flaskehals. Vi kan aftale målinger af kvalitet, gennemløb og gendannelse.

Se, hvad det koster Kontakt os →

Vil du have API og database til at følges ad uden besvær?

Spørg til beslutningerne bag »Har designet en JSON-passthrough-arkitektur«. I taler med mennesker, der kan forklare arbejdet og vurdere jeres eget projekt.

Se vores ydelser

welcome@engineer.company +45 50 10 61 84 København, Danmark

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

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