

Teorifrågor

1. Kan du förklara skillnaden mellan synkron och asynkron kod i Node.js?

I en synkron kod när en operation påbörjas stannar den först programmet och sen väntar tills operationen är klar innan den går till nästa uppgift/instruktion. T.ex program som filinläsning så stannar programmet och väntar på att fil inläsningen ska slutföras. Sådana kod kan vara tidsfördröjande särskilt när det gäller operationer som kräver mer tid att utföra.

I en asynkron kod när en operation påbörjas fortsätter programmet med andra uppgifter/instruktioner utan att behöva vänta på att den ena operationen måste vara slutförd. Detta är mer oftast praktiskt då det är mer tidseffektivt när det gäller operationer som kräver mer tid att slutföra.

2. Beskriv användningen av middleware i Express och ge exempel på några vanliga middleware.

Hanterar och bearbetar förfrågningar (req) innan de når slutdestinationen (route handler). Varje middleware får tillgång till förfrågan (req), svarobjekt (res) och en funktion (next) som används för att överföra kontrollen till nästa middleware i köen. Middleware i Express är som checkpoints för inkommande förfrågningar. Varje checkpoint kan göra saker som att logga information, hantera autentisering eller fel, och sedan släppa förfrågan vidare till nästa checkpoint.

3. Vilka fördelar erbjuder Prisma när det gäller att interagera med databasen?

- Det kan göra det enklare att hantera relationer mellan olika tabeller i databasen.
- Stödjer flera olika databaser såsom MySQL, PostgreSQL osv.
- Du kan byta databas utan att behöva ändra mycket i din kodbas.
- Är hyfsad enkel att komma igång med.
- Den kan generera SQL-frågor automatiskt baserat på ditt databasschema och dina prisma-queries.

4. Hur definierar du en modell för en tabell i Prisma?

Du definierar en modell för en tabell i Prisma i din databas genom att använda Prisma Schema Definition Language (SDL). Här är en exempel på hur du definierar en modell för en tabell i prisma:

1. Skapa en Prisma `schema.prisma`-fil i en mapp för prisma-filer
2. Definiera modellen för model User genom att använda Prisma SDL. Exempel finns nedanför:

```
// schema.prisma

model User {
  id Int @id @default(autoincrement())
  name String
  email String @unique
}
```

- **Model:** Definierar en Prisma-modell, och "User" är namnet på tabellen.
- **id:** Primärnyckel för tabellen med typen `Int`. **@id** indikerar att detta är primärnyckeln, och **@default(autoincrement())** betyder att värdet ökar automatiskt.
- **name:** En kolumn för användarens namn av typen `String`.
- **email:** En kolumn för användarens e-postadress av typen `String`, och **@unique:** indikerar att värdet måste vara unikt i hela tabellen.

5. Vad är skillnaden mellan `npm install` och `npm install --save`?

- **npm install** används för att installera paket (dependencies) som är specificerade i `package.json`-filen för ditt projekt.
- **npm install --save** användes för att installera ett npm-paket och samtidigt uppdatera dependencies-sektionen i `package.json` med den installerade paketversionen. Dock behövs inte denna kommando längre då det finns npm-versionen 5.0 som gör detta automatisk när du skriver `npm install`. Innan 5.0 kom då användes `--save` förut.