

BACKEND

Úvod do tvorby backendových služeb v Node.js. Knihovna express, opakování databází a jejich použití.

BACKEND

Backend je klíčová součást každé webové aplikace, která zpracovává data, komunikuje s databází a poskytuje tato data frontendu prostřednictvím API. Také zajišťuje bezpečnost a autentizaci uživatelů.

Technologie a Frameworky

Existuje mnoho technologií a frameworků, které se používají pro vývoj backendu. Mezi nejpopulárnější patří:

- **Node.js a Express**
- **PHP**
- **Ruby on Rails**
- **Django a Flask**
- **ASP.NET**
- **Java Spring**

Každý z těchto nástrojů má své výhody a je vhodný pro různé typy projektů.

Proč se k API nemůžeme připojovat z frontendu?

Připojování k API přímo z frontendu může představovat bezpečnostní riziko. Backendová vrstva poskytuje ochranu proti různým útokům, jako je SQL injection, a zajišťuje, že citlivé operace jsou chráněny před přímým přístupem.

Poznámky:

Žádné poznámky k této lekci jste ještě nezapsali.

HTTP

HTTP (Hypertext Transfer Protocol) je protokol pro přenos dat mezi klientem a serverem. Pracuje na principu požadavek-odpověď, kde klient posílá požadavek a server odpovídá. Požadavky i odpovědi mohou obsahovat hlavičky, které jsou páry klíč-hodnota, a tělo, které obsahuje data. HTTP používá TCP, ale v budoucnu přejde na HTTP/3, který bude používat UDP.

HTTP Požadavek

Každý HTTP požadavek má několik klíčových částí:

- **Metoda:** např. GET, POST, PUT, DELETE, PATCH, OPTIONS, HEAD
- **URL:** cesta bez protokolu a domény
- **Hlavičky:** dodatečné informace, které požadavek nese
- **Tělo:** data, která jsou posílána s požadavkem (u metod jako POST)

HTTP Odpověď

Odpověď z serveru obsahuje:

- **Kód odpovědi:** např. 200 (OK), 404 (Not Found), 500 (Internal Server Error)
- **Statusový text:** krátký popis kódu odpovědi
- **Hlavičky:** dodatečné informace
- **Tělo:** data, která server posílá zpět

Nejznámější Hlavičky

Některé z nejznámějších hlaviček zahrnují:

- `Content-Type` : typ obsahu (např. `application/json`)
- `Content-Length` : délka těla odpovědi
- `Authorization` : informace pro autentizaci
- `Cache-Control` : kontrola kešování
- `Cookie` a `Set-Cookie` : práce s cookies
- `User-Agent` : informace o klientovi
- `Referer` : URL odkazu
- `Host` : doména a port
- `Access-Control-Allow-Origin` : povolení CORS (Cross-Origin Resource Sharing)

REST API

REST (Representational State Transfer) je architektura pro vytváření API. Pracuje se zdroji, které jsou reprezentovány pomocí různých formátů (např. JSON). Přístup k datům je bezstavový, což znamená, že každý požadavek musí obsahovat veškeré informace potřebné k jeho vyřízení. REST API používá standardní HTTP metody (GET, POST, PUT, DELETE) pro operace se zdroji.

Poznámky:

Žádné poznámky k této lekci jste ještě nezapsali.

EXPRESS

Express je nejpopulárnější framework pro Node.js. Je jednoduchý na použití, ale poskytuje řešení pro různé problémy, jako je routování, middleware, a práce se šablonami. Express má také mnoho pluginů, které rozšiřují jeho funkčnost.

Instalace a Použití

Pro použití Express stačí nainstalovat balíček `express` do projektu, vytvořit instanci aplikace, a nastavit aplikaci, aby naslouchala na určitém portu. Tento port se pak používá v aplikacích pro komunikaci s backendem.

```
import express from 'express';

const app = express();

app.listen(3000, () => {
  console.log('Server is running on http://localhost:3000');
});
```

Middleware

Middleware v Expressu jsou funkce, které mají přístup k požadavku (req), odpovědi (res), a funkci next. Middleware může provést nějakou akci a předat požadavek dalšímu middleware v pořadí. Middleware lze použít globálně pro celou aplikaci, nebo jen pro konkrétní endpointy.

```
app.use((req, res, next) => {
  console.log('Request received');
  next();
});
```

Middleware může být také použito pro omezení přístupu, práci s CORS, a zpracování vstupních dat.

Vstup

Vstupní data mohou být čtena z těla požadavku, URL (query parametry a cesta), nebo hlaviček. Pro zpracování těchto dat se často používá middleware jako `body-parser`, který umožňuje jednoduše pracovat s různými formáty dat.

Validace

Pro validaci vstupních dat se používají knihovny jako `express-validator` nebo `joi`. Validace je důležitá, protože vstupní data mohou být nedůvěryhodná a mohou obsahovat chyby nebo škodlivý kód.

Poznámky:

Žádné poznámky k této lekci jste ještě nezapsali.

DATABÁZE

Existují různé typy databází, jako jsou relační databáze (SQL) a dokumentové databáze (NoSQL). Relační databáze ukládají data v tabulkách, které mají řádky a sloupce. Každá tabulka má primární klíč, který jednoznačně identifikuje každý řádek, a cizí klíče, které odkazují na jiné tabulky.

SQL

SQL (Structured Query Language) je jazyk pro práci s relačními databázemi. Pomocí SQL můžeme vytvářet tabulky, vkládat data, vybírat data, aktualizovat data, a mazat data.

Použití DB na Backend

V Node.js je mnoho knihoven pro práci s databázemi, jako je `pg` pro PostgreSQL nebo ORM (Object-Relational Mapping) nástroje jako `sequelize` a `prisma`. Tyto nástroje usnadňují práci s databázemi a poskytují bezpečné způsoby pro práci s daty.

Připojení k DB

```
const { Client } = require('pg');
const client = new Client({
  connectionString: 'postgresql://user:password@localhost:5432/mydb'
});
client.connect();
```

Výběr Dat

```
client.query('SELECT * FROM users', (err, res) => {
  console.log(res.rows);
  client.end();
});
```

SQL Injection

SQL Injection je bezpečnostní chyba, kdy útočník může vložit vlastní SQL kód do parametru. Tomu se lze vyhnout použitím správných metod pro práci s databází, jako jsou prepared statements.

```
const text = 'SELECT * FROM users WHERE id = $1';
const values = [1];
client.query(text, values, (err, res) => {
  console.log(res.rows);
  client.end();
});
```

Poznámky:

Žádné poznámky k této lekci jste ještě nezapsali.

REVERZÍ PROXY

HTTP používá port 80 a HTTPS port 443. Na jednom PC může běžet více serverů, pokud používají různé porty. Reverzní proxy je server, který přesměrovává požadavky na jiný server nebo port, což se často používá pro zajištění bezpečnosti, load balancingu a škálování.

Poznámky:

Žádné poznámky k této lekci jste ještě nezapsali.

