

Webové technologie a vývoj (Fullstack)

Moderní webový vývoj se dělí na dvě hlavní oblasti: **Frontend** (klientská část) a **Backend** (serverová část). Jejich propojením vzniká **Fullstack** vývoj. Cílem je vytvořit aplikaci, která je rychlá, bezpečná a škálovatelná.

1. Frontend: Moderní frameworky

S rostoucí komplexitou webů už nestačí čistý JavaScript. Vývojáři používají frameworky, které umožňují stavět weby z opakovaně použitelných komponent.

- **React (Meta)**: Knihovna založená na komponentách a virtuálním DOMu. Aktuálně nejpoblárnější volba.
- **Vue.js**: Přístupnější framework s velmi čistou dokumentací, oblíbený pro svou flexibilitu.
- **Angular (Google)**: Robustní framework (vše v jednom), vhodný pro velké podnikové aplikace.
- **Svelte**: Moderní přístup, který kompiluje kód do vysoce efektivního čistého JS již při sestavení.

2. Backend: Srdce aplikace

Backend zpracovává logiku, autentizaci uživatelů a komunikaci s databázemi.

Populární technologie:

- **Node.js**: Umožňuje psát backend v JavaScriptu. Je extrémně rychlý díky asynchronnímu I/O.
- **Python (Django, Flask)**: Skvělý pro rychlý vývoj a aplikace pracující s daty.
- **PHP (Laravel)**: Stále velmi rozšířený jazyk, který pohání velkou část webu (např. WordPress).
- **Java (Spring)**: Standard pro vysoce bezpečné a rozsáhlé systémy (bankovníctví).

3. Komunikace: API a protokoly

Frontend a Backend spolu mluví pomocí rozhraní **API** (Application Programming Interface).

- **REST API**: Nejrozšířenější standard využívající HTTP metody (GET, POST, PUT, DELETE) a formát **JSON**.
- **GraphQL**: Moderní alternativa od Facebooku, která umožňuje klientovi říct si přesně o ta data, která potřebuje (šetří přenesená data).
- **WebSockets**: Technologie pro obousměrnou komunikaci v reálném čase (např. chaty, burzovní grafy).

4. Databáze na webu

Webové aplikace musí data někde trvale uchovávat.

- **Relační (SQL):** Data jsou v tabulkách s jasnými vztahy. (PostgreSQL, MySQL).
- **Nerelační (NoSQL):** Data jsou v dokumentech (JSON), vhodné pro nestrukturovaná data a vysoký výkon. (MongoDB, Redis).

5. Nasazení a infrastruktura (Deployment)

Hotový kód musí běžet na serveru. Moderní přístup využívá:

- **Kontejnery (Docker):** Aplikace je zabalena se všemi závislostmi, takže běží všude stejně.
- **CI/CD:** Automatické nasazování. Jakmile vývojář pošle kód do [Gitu](#), spustí se [testy](#) a aplikace se sama aktualizuje na serveru.
- **Serverless:** Vývojář neřeší servery, pouze nahrává kód, který se spustí při požadavku (AWS Lambda, Vercel).

6. Webová bezpečnost

Vývojář musí chránit aplikaci před útoky:

- **HTTPS:** Šifrování přenosu pomocí TLS certifikátu.
- **XSS (Cross-Site Scripting):** Zabránění vložení cizího škodlivého skriptu do stránky.
- **SQL Injection:** Ochrana proti podvržení příkazů do databázového dotazu.

Související články:

- [Základy webu \(HTML, CSS, JS\)](#)
- [Verzování kódu \(Git\)](#)
- [Cloud computing \(AWS, Azure, GCP\)](#)

Tagy: dev web frontend backend fullstack react nodejs api sql

From:

<https://serviceit.cz/> - IT ENCYKLOPEDIÉ

Permanent link:

https://serviceit.cz/doku.php?id=it:dev:web_tech

Last update: **2026/01/02 13:43**



