

GraphQL

GraphQL vyvinul Facebook v roce 2012 (open-source od roku 2015) jako řešení problémů s efektivitou mobilních aplikací. Je to silně typovaný jazyk, který umožňuje klientům získat všechna potřebná data v rámci jediného síťového požadavku.

1. Hlavní rozdíly oproti RESTu

V tradičním RESTu musíte často volat několik adres (např. `/uzivatel/1` a pak `/uzivatel/1/prispevky`), abyste získali kompletní data. GraphQL to mění:

- **Jeden koncový bod (Endpoint):** Všechny dotazy směřují na jednu URL (typicky `/graphql`).
- **Žádné Over-fetching:** Klient nedostává zbytečná data, která nepoužije.
- **Žádné Under-fetching:** Klient získá vše v jednom dotazu, nemusí posílat další požadavky pro související data.

2. Tři pilíře GraphQL

Provoz GraphQL stojí na třech základních operacích:

- **Query (Dotaz):** Používá se pro čtení dat (ekvivalent GET v RESTu).
- **Mutation (Mutace):** Používá se pro zápis, úpravu nebo mazání dat (ekvivalent POST, PUT, DELETE).
- **Subscription (Odběr):** Umožňuje serveru posílat data klientovi v reálném čase přes WebSockets (např. pro chat).

3. Schema a Typový systém

Srdcem každého GraphQL API je **Schema**, které definuje, jaká data jsou dostupná. Používá k tomu jazyk SDL (Schema Definition Language).

```
# Definice typu na serveru
type Uzivatel {
  id: ID!
  jmeno: String!
  email: String
  prispevky: [Prispevek]
}

# Dotaz od klienta
```

```
query {  
  uživatel(id: "42") {  
    jmeno  
    email  
  }  
}
```

4. Resolvery (Zpracování dat)

GraphQL neříká, jak máte data ukládat. Na serveru existují funkce zvané **Resolvery**, které mají za úkol „vyřešit“ (získat) konkrétní pole dotazu. Jeden resolver může sahat do SQL databáze, druhý do NoSQL a třetí může volat úplně jiné API.

5. Výhody a nevýhody

Výhody	Nevýhody
Flexibilita: Klient si říká o strukturu dat.	Komplexita: Náročnější implementace na straně serveru.
Typová bezpečnost: Automatická dokumentace a validace.	Cachování: Je složitější než u RESTu (HTTP cache zde nefunguje).
Verzování: Není potřeba `/v1/`, `/v2/` – stačí přidávat nová pole.	N-plus-one problém: Riziko neefektivních databázových dotazů.

6. Populární nástroje

- **Apollo Server/Client:** Nejrozšířenější ekosystém pro JavaScript/TypeScript.
- **HotChocolate:** Špičková implementace pro [.NET](#).
- **GraphiQL / Playground:** Interaktivní prohlížeč pro testování dotazů přímo v prohlížeči.

Související články:

- [REST API: Klasický přístup](#)
- [Formát JSON](#)
- [Webový vývoj v .NET](#)

Tagy: *programming api graphql facebook web-development backend query-language*

From:
<https://www.serviceit.cz/> - **IT ENCYKLOPEDIE**

Permanent link:
<https://www.serviceit.cz/doku.php?id=it:sw:graphql>

Last update: **2026/01/02 19:56**

