REST (Representational State Transfer)

REST je architektonický styl definující sadu principů pro vytváření webových služeb. Služby odpovídající těmto principům se označují jako **RESTful**. REST není protokol (jako SOAP), ale využívá existující protokol **HTTP**.

Základní myšlenkou RESTu je práce se **zdroji** (resources), které jsou identifikovány pomocí unikátních adres (URI).

Základní principy RESTu

Aby bylo rozhraní považováno za RESTful, mělo by splňovat tyto podmínky:

- **Bezstavovost (Statelessness):** Server neukládá žádné informace o relaci klienta. Každý požadavek musí obsahovat všechna data potřebná k jeho zpracování.
- **Oddělení klienta a serveru:** Klient se stará o uživatelské rozhraní, server o data a logiku. Mohou se vyvíjet nezávisle.
- **Jednotné rozhraní:** Používání standardních HTTP metod a formátů (nejčastěji JSON nebo XML).
- **Možnost kešování:** Odpovědi musí definovat, zda je možné je uložit do mezipaměti, aby se snížila zátěž sítě.

HTTP Metody v RESTu

REST využívá standardní slovesa protokolu HTTP k provádění operací nad zdroji (často přirovnávané k operacím CRUD):

HTTP Metoda	CRUD operace	Význam
GET	Read	Získání dat o zdroji (nemění stav).
POST	Create	Vytvoření nového zdroje.
PUT	Update/Replace	Úplná aktualizace (nahrazení) zdroje.
PATCH	Update/Modify	Částečná aktualizace zdroje.
DELETE	Delete	Odstranění zdroje.

URI a Struktura zdrojů

Zdroje jsou v RESTu reprezentovány pomocí podstatných jmen v množném čísle.

Příklady volání API:

- `GET /uzivatele` – Získá seznam všech uživatelů.
- `GET /uzivatele/42` – Získá detail uživatele s ID 42.
- `POST /uzivatele` – Vytvoří nového uživatele (data jsou v těle požadavku).
- `GET /uzivatele/42/objednavky` – Získá objednávky konkrétního uživatele.

Odpovědi a stavové kódy

Server vrací data (obvykle ve formátu JSON) společně s **HTTP stavovým kódem**, který indikuje výsledek operace.

Kód	Význam	Příklad použití
200 OK	Úspěch	Data byla v pořádku doručena.
201 Created	Vytvořeno	Nový zdroj byl úspěšně vytvořen (po POST).
400 Bad Request	Chyba klienta	Neplatná syntaxe požadavku.
401 Unauthorized	Neautorizováno	Chybí nebo je neplatné přihlášení.
404 Not Found	Nenalezeno	Požadované URI neexistuje.
500 Internal Server Error	Chyba serveru	Neočekávaná chyba na straně serveru.

Formáty dat

Ačkoliv REST může teoreticky přenášet jakýkoliv formát, v moderní praxi dominuje **JSON** díky své lehkosti a snadné čitelnosti v JavaScriptu.

```
// Příklad odpovědi v JSON
{
  "id": 42,
  "jmeno": "Jan Novák",
  "email": "jan.novak@example.com",
  "role": ["admin", "uzivatel"]
}
```

HATEOAS (Pokročilý koncept)

Hypermedia as the Engine of Application State je princip, kdy server v odpovědi posílá i odkazy na další možné akce, takže klient nemusí mít adresy „nadrátované“ napevno.

> **Tip:** Pro testování a dokumentaci REST API se dnes jako standard používá nástroj **Swagger (OpenAPI)** nebo aplikace **Postman**.

Zpět na Data