



Informační systémy
na míru



Business
analytika



Mobilní
aplikace



Weby
a E-shopy

{ REST API }



1. MÍSTO

IBM

FIRMA ROKU 2021

PLZEŇSKÉHO KRAJE

RTsoft s.r.o.

- Plzeňská softwarová společnost působící na trhu již přes 20 let
- Vývoj webových a mobilních aplikací na míru
- Více než 300 dokončených projektů
- Mezi naše partnery patří MND, Knihy Dobrovský (3× e-shop roku), Sazka, Sportisimo, Darujme.cz a mnoho dalších
- Více než 40 vývojářů
- Přes e-shopy našich zákazníků proteklo za 2021 přes 2 mld. Kč

RTsoft s.r.o.



API a REST API

API

- API = Application Programming Interface
- rozhraní pro komunikaci mezi aplikacemi
- client x server, request x response

REST API vs HTTP

- REST - sada pravidel
- HTTP - komunikační protokol

REST API

- Client - Server
- Uniform Interface
- Stateless
- Cacheable
- Layered architecture

Client - Server

- nezávislost klienta na serveru a obráceně
- jednoduchá škálovatelnost na obou stranách

Uniform Interface

- jasná identifikace jednotlivých resources (URI)
- možnost modifikace resources (POST, PUT, PATCH)
- součástí response je informace o typu dat (Content-Type)
- HATEOAS



HATEOAS

GET <https://www.awesome-shop.com/orders/5>

```
{
  "order": {
    "id": 5,
    "state": "new",
    "total_amount": 10000,
    "links": [
      {
        "href": "5/products",
        "rel": "order_products",
        "type": "GET"
      },
      {
        "href": "5/cancel",
        "rel": "cancel",
        "type": "PUT"
      }
    ]
  }
}
```

HATEOAS

GET <https://www.awesome-shop.com/products?page=4>

```
{
  "page": 4,
  "total_pages": 50,
  "total_count": 2000,
  "products": [
    {
      "id": 1,
      "name": "Brand new super product",
    },
  ],
  "links": [
    {
      "href": "products?page=5",
      "rel": "next_page",
      "type": "GET"
    },
    {
      "href": "products?page=3",
      "rel": "prev_page",
      "type": "GET"
    }
  ]
}
```

Stateless

- nezávislost requestů na stavu uloženém na serveru
- každý request si s sebou nese potřebná data
- např. autentizace - místo session použití token based autentizace

Cacheable

- možnost cache API response na HTTP vrstvě
- snížení loadu na webový server, rychlejší odezvy

Layered System

- client nezná implementační detaily serveru
- proxy / gateway
- security layer, load balancing layer, cache layer



API DESIGN

API Use Cases

- API mobilní aplikace
- API pro SPA nebo PWA
- Uživatelské API

CRUD vs Intent

- CRUD based (create, read, update, delete)
- intent based
 - POST /order-cancelation
 - POST /cash-transfer
 - POST /product-shipment

HTTP metody

- GET
- POST
- PUT
- PATCH
- DELETE

PUT vs PATCH

PUT <https://www.awesome-shop.com/orders/1>

Payload:

```
{  
  "state": "shipped",  
  "total_amount": 10000  
}
```

PUT vs PATCH

PATCH <https://www.awesome-shop.com/orders/1>

Payload:

```
{  
  "state": "shipped"  
}
```

Idempotence HTTP metod

- idempotentní - GET, PUT, DELETE
- ne-idempotentní - POST, PATCH

CRUD - URIs

- podstatná jména pro URI v množném čísle
- GET /orders
- GET /orders/10

CRUD - URIs

- GET /orders/10/order-items
- GET /orders/code/XYZ
- GET /orders?email=hora@rtsoft.cz

CRUD - URIs

- POST /orders
- PUT /orders/10
- PATCH /orders/10
- DELETE /orders/10

API RESPONSE CODES

API Response Codes

- slouží k detailnímu rozlišení stavu odpovědi ze serveru
- 2xx
- 4xx
- 5xx

2xx

- jen jediný stav by měl být 200 OK
- 200 OK
- 201 Created
- 204 No Content

4xx

- 400 Bad Request
- 401 Unauthorised
- 403 Forbidden
- 404 Not Found
- 422 Unprocessable Entity

5xx

- 500 Server Error
- 501 Not Implemented
- 502 Bad Gateway
- 503 Service Unavailable
- 504 Gateway Timeout

400 Bad Request

POST /product

Payload:

```
{  
  "in_stock": 1000,  
  "active": true  
}
```

400 response

```
{  
  "error": "Invalid data shape",  
  "errorDetail": "Property name missing"  
}
```

422

POST /product

Payload:

```
{  
  "name": "",  
  "in_stock": 1000,  
  "active": true  
}
```

422 response

```
{  
  "error": "Invalid input",  
  "errorDetail": "Property name is required"  
}
```

RESOURCES

Resource

- entita nebo kolekce vrácena v response nebo odeslána v requestu
- struktura resource != DB struktura
- reprezentace resource - JSON, XML, ...

Resource

GET <https://www.awesome-shop.com/orders/1>

```
{  
  "id": 1,  
  "state": "shipped",  
  "total_amount": 10000  
}
```

GET https://www.awesome-shop.com/orders/1

```
{
  "id": 1,
  "state": "shipped",
  "total_amount": 10000,
  "customer": {
    "name": "Jane Doe",
    "email": "jane@doe.com",
    "order_counts": 10
  },
  "order_items": [
    {
      "id": 1,
      "name": "Mega super product",
      "long_name": "Giga mega super product",
      "price": 9500
    },
    {
      "id": 2,
      "name": "Shipping & Payment",
      "long_name": "Shipping & Payment",
      "price": 500
    }
  ]
}
```

GET https://www.awesome-shop.com/orders/1?include=customer,order_items

```
{
  "id": 1,
  "state": "shipped",
  "total_amount": 10000,
  "customer": {
    "name": "Jane Doe",
    "email": "jane@doe.com",
    "order_counts": 10
  },
  "order_items": [
    {
      "id": 1,
      "name": "Mega super product",
      "long_name": "Giga mega super product",
      "price": 9500
    },
    {
      "id": 2,
      "name": "Shipping & Payment",
      "long_name": "Shipping & Payment",
      "price": 500
    }
  ]
}
```

GET <https://www.awesome-shop.com/order-items/2>

```
{  
  "id": 2,  
  "name": "Shipping & Payment",  
  "long_name": "Shipping & Payment",  
  "price": 500  
}
```

INTENT BASED API

Intent based API

- vhodné použití tam, kde se CRUD based architektura nehodí
- např. situace, kdy by se v CRUD based modelu by se musely posílat více separátních requestů

Intent based API

- schválení objednávky
 - označení objednávky příznakem schváleno
 - vyskladnění produktů

Intent based API

- převod financí z účtu na účet
 - PUT request na odebrání peněz z jednoho účtu
 - PUT request na přidání peněz na druhý účet

Intent based API

POST /order-approval/5

Intent based API

POST /transfer

```
{  
  "sourceAccountId": 5,  
  "targetAccountId": 7,  
  "amount" : 1000  
}
```

Dokumentace API

- Swagger (OpenAPI)
- Swagger PHP (<https://github.com/zircote/swagger-php>)
- Apiary (API Blueprint)

Stateless autentizace

- JWT token
- Authorization: Bearer <token>
- ověřovat identitu uživatele na serveru
- secret, expiration

Verzování API

- URL - `https://www.awesome-shop.com/v2/orders/1`
- Custom header - `Accepts-version: 2.0`
- Accept header - `Accept: application/json; version=2`

Benefity dodržování REST API best practices

- čistý a předvídatelný design -> rychlejší onboarding vývojářů
- lepší flexibilita a rozšiřitelnost API
- škálovatelnost výkonu

Co si zapamatovat

- CRUD x Intent
- Být konzistentní (URI, ...)
- HTTP response status kódy
- Dokumentace

Stále nabíráme!

www.rtsoft.cz/kariera

Logujeme



29. 6. 2022

18:00

**Restaurace U Salzmannů
salonek v 1. patře**

občerstvení zajištěno
vstup zdarma

