

OBJEKTIVÉ MODELOVÁNÍ, B9B36OMO

GRAPHQL VS REST API

JIŘÍ ŠEBEK

Katedra počítačů
Fakulta elektrotechnická
České vysoké učení technické v Praze

CO JE REST API?



CO JE REST API?

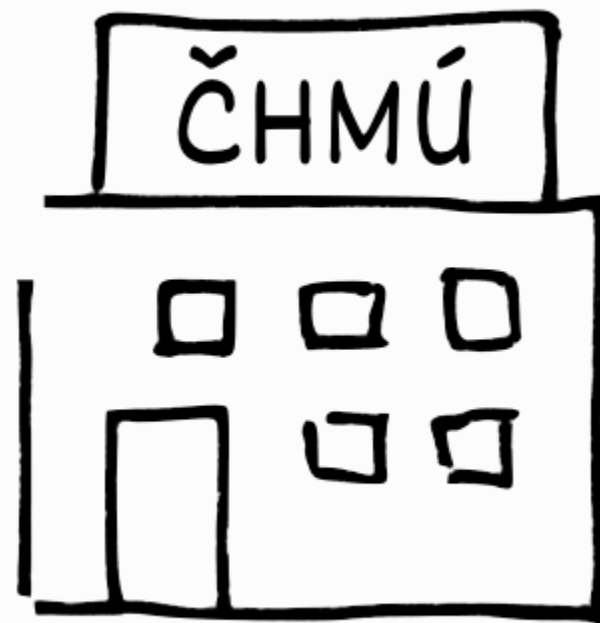


- ARCHITEKTURA, KTERÁ UMOŽŇUJE PŘISTUPOVAT K DATŮM POMOCÍ HTTP
- JE ORIENTOVÁNA DATOVĚ, NIKOLI PROCEDURÁLNĚ (OPROTI SOAP, CORBA...)
- METODY PRO PŘÍSTUP KE ZDROJŮM
 - GET, PUT, POST, DELETE

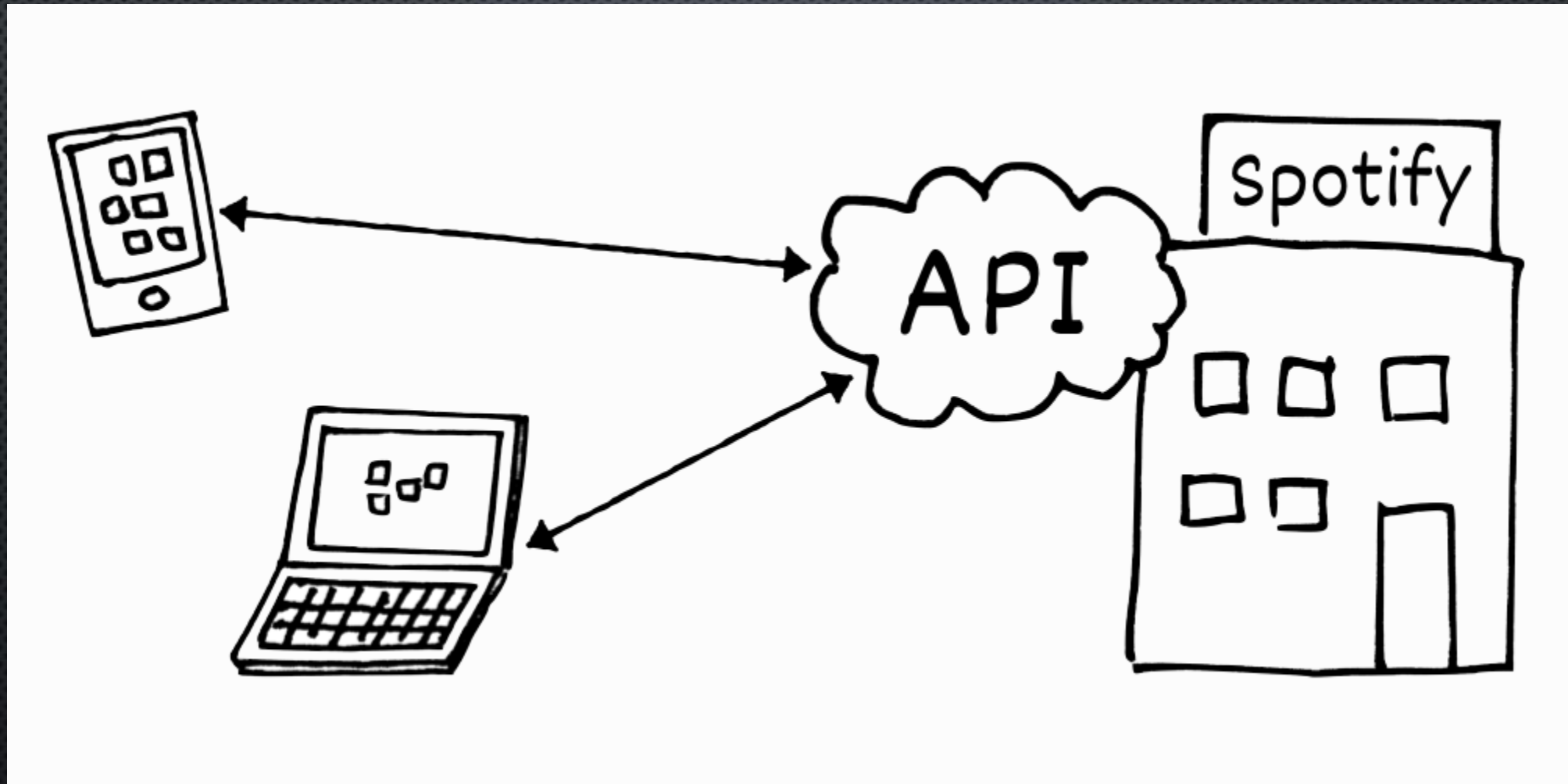
A CO JE JEN API?



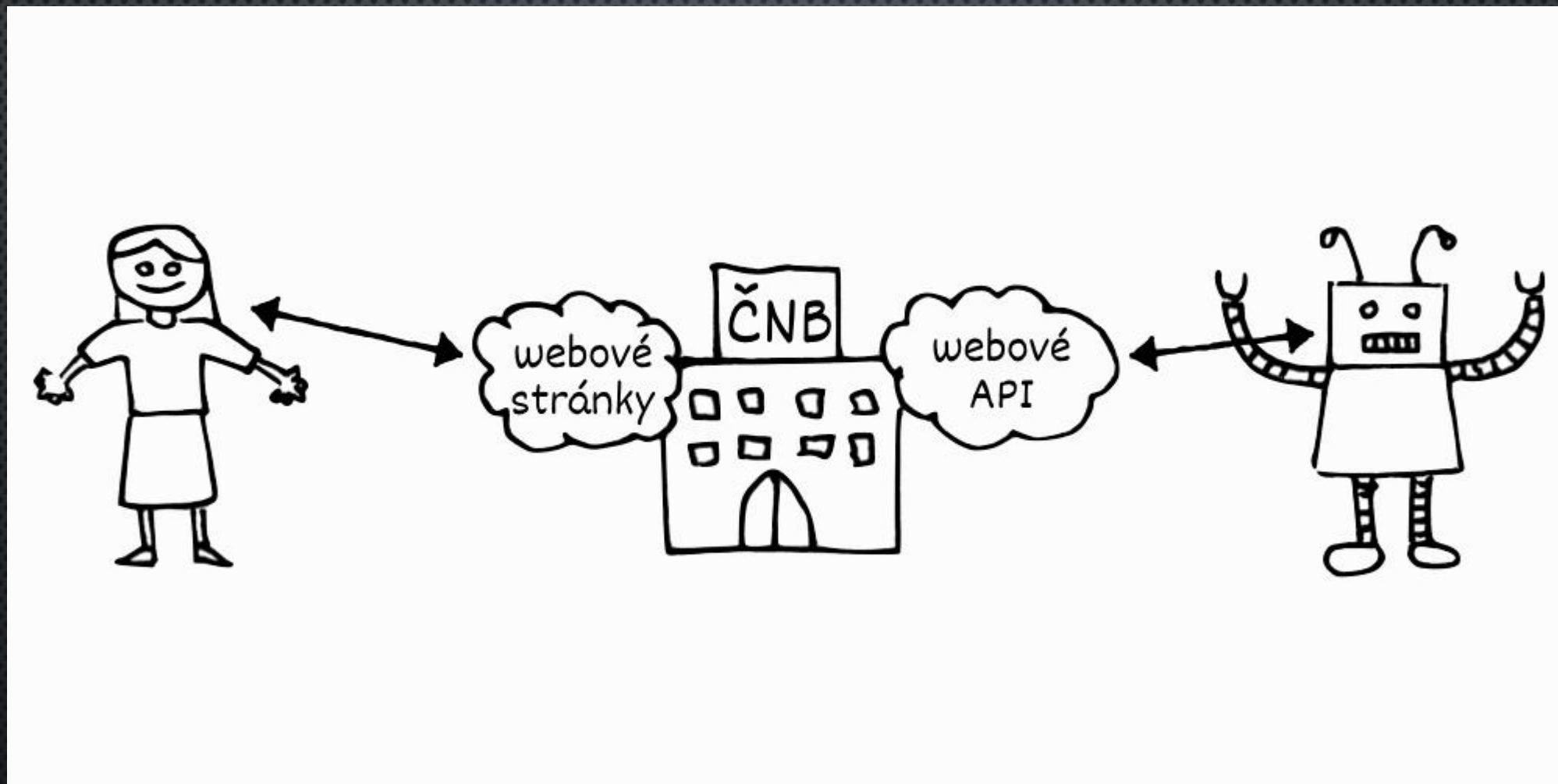
???



A CO JE JEN API?



A CO JE JEN API?



Rest API Basics

Typical HTTP Verbs:

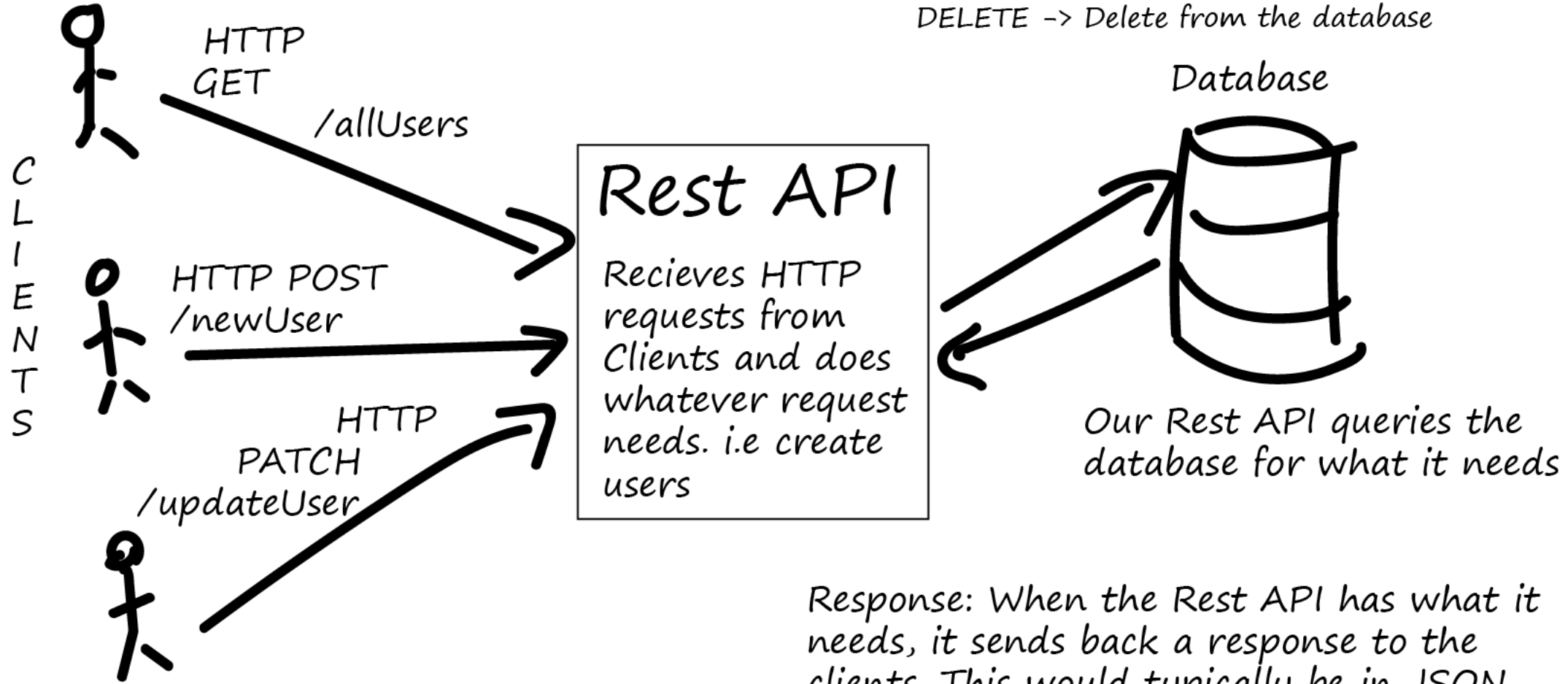
GET -> Read from Database

PUT -> Update/Replace row in Database

PATCH -> Update/Modify row in Database

POST -> Create a new record in the database

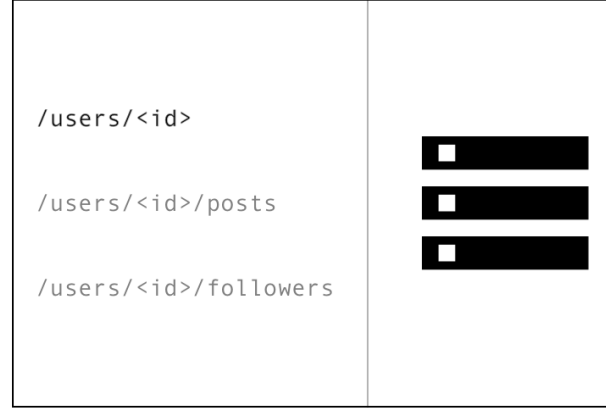
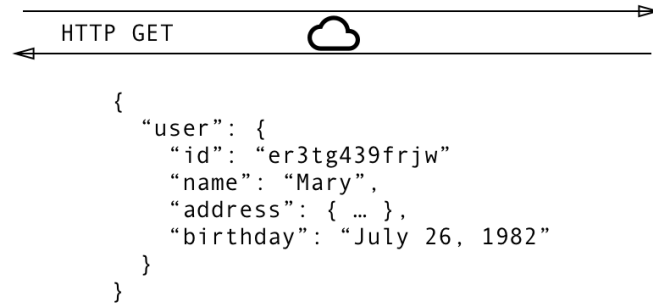
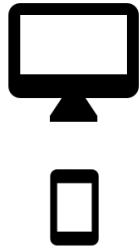
DELETE -> Delete from the database



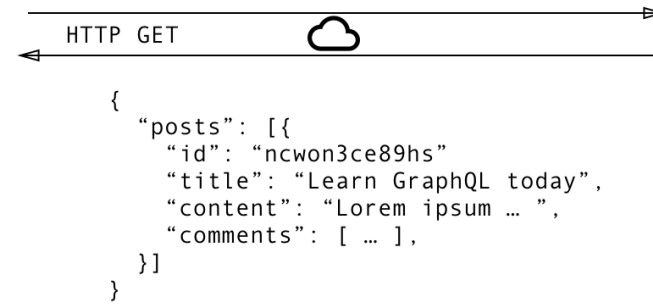
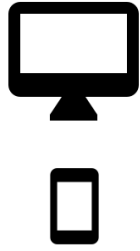
Our Clients, send HTTP Requests and wait for responses

Response: When the Rest API has what it needs, it sends back a response to the clients. This would typically be in JSON or XML format.

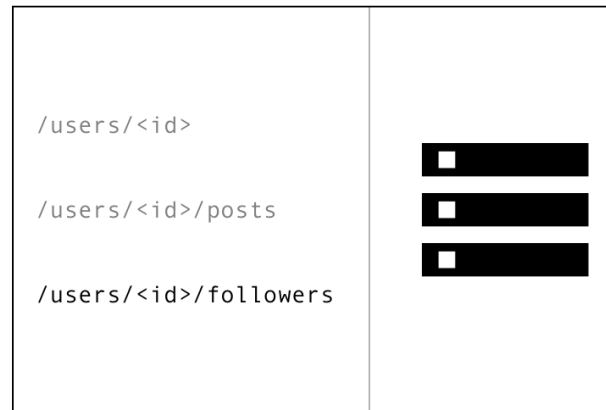
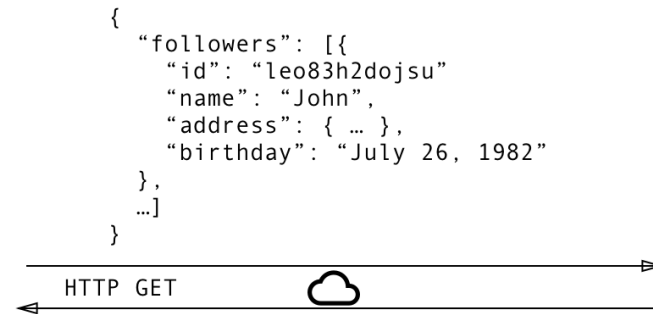
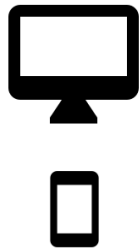
1



2



3



Insomnia

GET base_url /employees

Send

200

TIME 124 ms

SIZE 2.2 KB

9 Months Ago

local

Cookies

Body

{local}: localhost:8080

Auth

Query

Header

Docs

Preview

Header 3

Cookie

Timeline

Filter

+

GET GET employees

POST POST

token	1111	<input type="checkbox"/>	<input type="checkbox"/>
header	value	<input type="checkbox"/>	<input type="checkbox"/>
New header	New value		

```
1+ {
2+   "_embedded": {
3+     "employees": [
4+       {
5+         "id": 2,
6+         "firstName": "Bilbo",
7+         "lastName": "Baggins",
8+         "role": "burglar",
9+         "_links": {
10+           "self": {
11+             "href": "http://localhost:8080/employees/2"
12+           },
13+           "employees": {
14+             "href": "http://localhost:8080/employees"
15+           }
16+         },
17+         "_templates": {
18+           "default": {
19+             "title": null,
20+             "method": "put",
21+             "contentType": "",
22+             "properties": [
23+               {
24+                 "name": "firstName",
25+                 "required": true
26+               },
27+               {
28+                 "name": "id",
29+                 "required": true
30+               },
31+               {
32+                 "name": "lastName",
33+                 "required": true
34+               },
35+               {
36+                 "name": "role",
37+                 "required": true
38+               }
39+             ]
40+           }
41+         }
42+       }
43+     ]
44+   }
45+ }
```

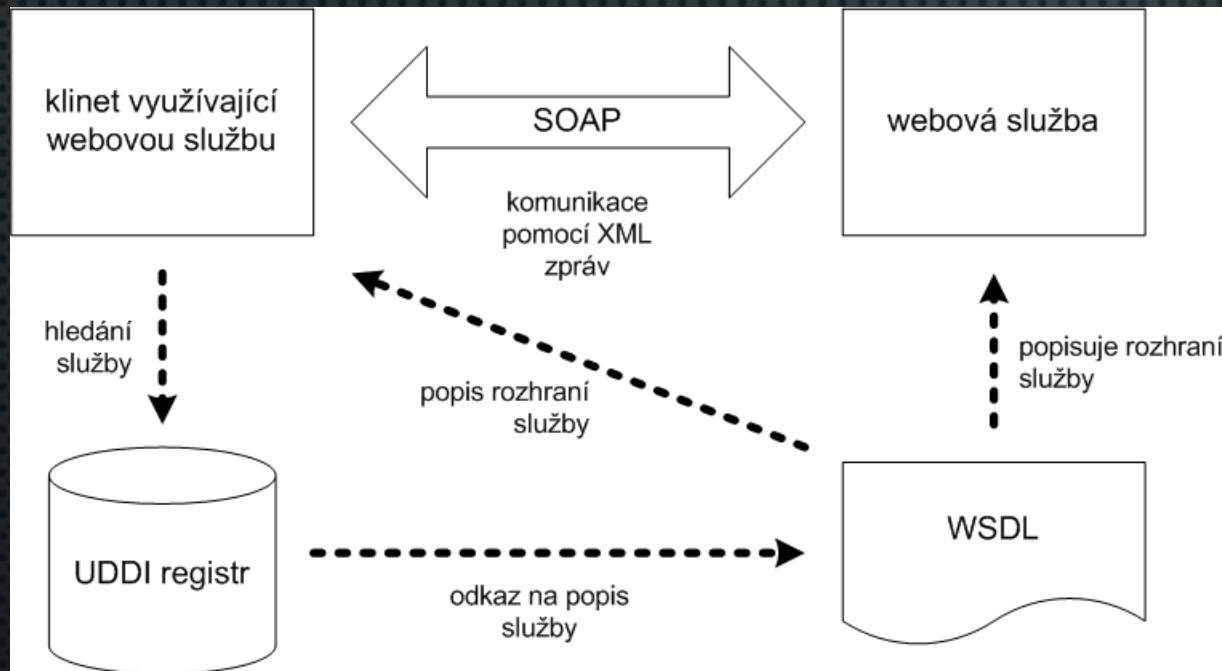
Bulk Edit

\$.store.books[*].author



ODBOČKA – CO JE SOAP?

- JE PROTOKOLEM PRO VÝMĚNU ZPRÁV ZALOŽENÝCH NA XML
- NÁSTUPCE XML-RPC



Request

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetails xmlns="http://warehouse.example.com/ws">
      <productID>827635</productID>
    </getProductDetails>
  </soap:Body>
</soap:Envelope>
```

Response

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetailsResponse
xmlns="http://warehouse.example.com/ws">
      <getProductDetailsResult>
        <productName>Čokoláda, sada 3 chutí</productName>
        <productID>827635</productID>
        <description>Čokoláda hořká, bílá a smetanová</description>
        <price>98,50</price>
        <inStock>ano</inStock>
      </getProductDetailsResult>
    </getProductDetailsResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP vs REST?

SOAP

1. Is a protocol through which two computer communicates by sharing XML document
2. Used only XML
3. Based reads cannot be cached
4. Is like custom desktop application, closely connected to the server
5. Is slower than REST

REST

1. It runs on HTTP
2. Rest is a service architecture and design for network-based software architectures
3. supports many different data formats
4. reads can be cached
5. A client is more like a browser; it knows how to standardized methods and an application has to fit inside it
6. Is faster than SOAP
7. It uses the HTTP headers to hold meta information

SOAP vs REST?

YOU STILL USE SOAP?



BETTER TAKE A REST

SOAP

1. Is a protocol through which sharing XML document
2. Used only XML
3. Based reads cannot be
4. Is like custom desktop a server
5. Is slower than REST



6. Is faster than SOAP
7. It uses the HTTP headers to hold meta information

CO JE GRAPHQL?



- JE VELMI EFEKTIVNÍM NÁSTUPCEM REST API ZAVEDENÝ FACEBOOKEM
- POUŽÍVANÝ GITHUBEM, PINTEREST A SHOPIFY
- GRAPHQL JE DOTAZOVACÍ JAZYK NA ÚROVNI **APLIKAČNÍ** VRSTVY PRO **API**
- DÁVÁ KLIENTSKÝM APLIKACÍM MOŽNOST VYŽÁDAT SI PŘESNĚ TA DATA, KTERÁ POTŘEBUJÍ A NIC NAVÍC

CO JE GRAPHQL?



GraphQL

Describe your data

```
type Project {  
  name: String  
  tagline: String  
  contributors: [User]  
}
```

Ask for what you want

```
{  
  project(name: "GraphQL") {  
    tagline  
  }  
}
```

Get predictable results

```
{  
  "project": {  
    "tagline": "A query language for APIs"  
  }  
}
```

GRAPHQL SPECIFICKÉ REQUESTY

```
{  
  hero {  
    name  
  }  
}
```

```
{  
  "hero": {  
    "name": "Luke Skywalker"  
  }  
}
```

```
{  
  hero {  
    name  
    height  
    mass  
  }  
}
```

```
{  
  "hero": {  
    "name": "Luke Skywalker",  
    "height": 1.72,  
    "mass": 77  
  }  
}
```

GRAPHQL TYPY

```
{
  hero {
    name
    friends {
      name
      homeWorld {
        name
        climate
      }
      species {
        name
        lifespan
        origin {
          name
        }
      }
    }
  }
}
```

```
type Query {
  hero: Character
}

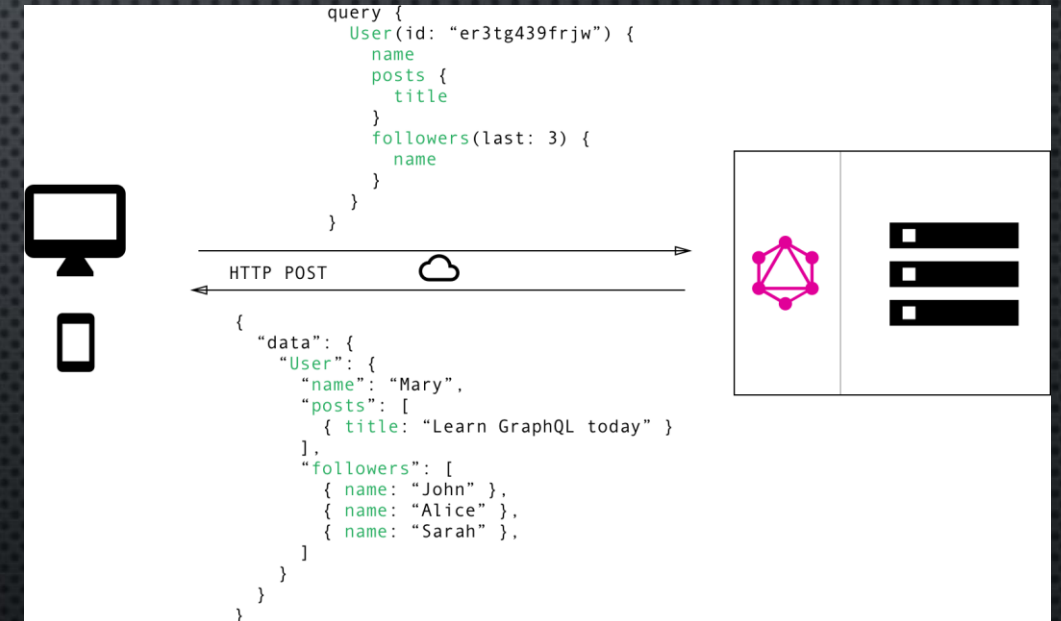
type Character {
  name: String
  friends: [Character]
  homeWorld: Planet
  species: Species
}

type Planet {
  name: String
  climate: String
}

type Species {
  name: String
  lifespan: Int
  origin: Planet
}
```


REST or ?
GraphQL

POROVNÁNÍ REST A GRAPHQL



GRAPHQL VS REST (TESTOVÁNÍ VÝKONU API)

Zatížení	Nízké		Střední		Vysoké		
Počet připojení	1		10		100		
Celkem dotazů	10		100		1000		
	APDEX	Průměr	APDEX	Průměr	APDEX	Průměr	Propustnost
GraphQL	0.950	118 ms	0.975	290 ms	0.373	1205 ms	49.76 r/s
REST	0.750	496 ms	0.435	1254 ms	0.003	12771 ms	6.98 r/s

APDEX vzorec:

$$\frac{s + \frac{t}{2}}{c}$$

S - udává počet požadavků, které byly provedeny v uspokojivém čase, což je v tomto případě 500 ms

t - udává počet požadavků splněných v tolerovaném čase, který je ve výpočtu nižší než 1 500 ms

c - udává celkový počet provedených požadavků

GRAPHQL VS REST (TESTOVÁNÍ VÝKONU API 2)

Typ API	Průměr	Počet požadavků na API
REST	4694.67 ms	15
GraphQL	2827.68 ms	6

ZÁVĚR



REST API



GraphQL

- EFEKTIVNĚJŠÍ KOMUNIKACI KLIENTA SE SERVEREM MÁ GRAPHQL (+ ZRYCHLIT PRVOTNÍ NAČTENÍ APLIKACE CCA 40%)
- GRAPHQL ZVLÁDÁ LÉPE VYŠŠÍ ZÁTĚŽ
- ..