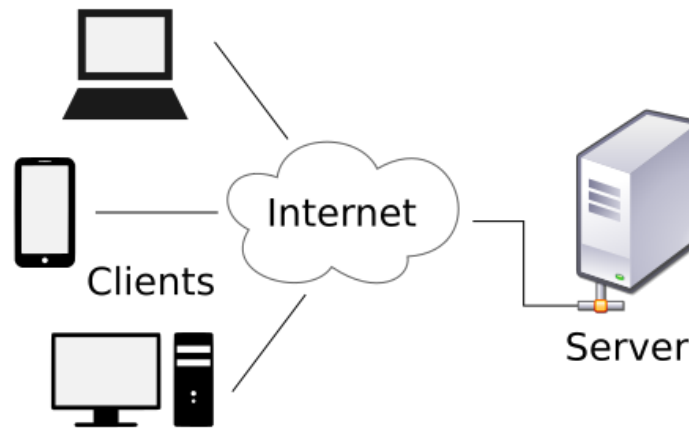


Client ↔ Server

Petr Čaněk

Client ↔ Server



- Server - any remote computer
- Client - any web, desktop, mobile application
- Communication is via Internet

**How do I get the
data?**

Pooling

- Client asks, Server responds
- Connection is short-lived

**I need to get
some files...**

Direct file access

- Only HTTP server needed
- Apache / Nginx ...
- No modifications / additions
- Can provide basic authentication

**What about
adding files?**

FTP

- File Transfer Protocol
- FTP server needed
- Works with files / folders only
- Enables CRUD operations
- Modification of file content must be done externally (editors)

CRUD operations

Create
Read
Update
Delete

**Well, I would like
to search and
filter the data...**

API

Application Programming Interface

API

- Custom written application
- Provides data and/or functionality
- Can be of any complexity and shape

**I would like some
structure for my
API...**

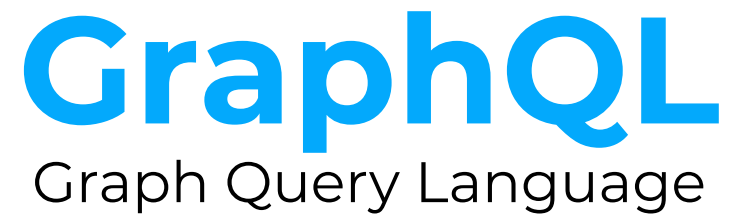
REST

Representational State Transfer

REST

- Oriented around resources
- One resource == one URL
- <https://domain.io/api/v2/products/?category=3&material=2,3>
- Leverages standard HTTP methods
- GET, POST, PUT, PATCH, DELETE
- Is "weak" standard
- Resource is server "as-is"
- Every implementation is different

```
▼ {
  "num_pages": 54,
  "count": 484,
  "next": "https://staging.admin-api.prettique.de/api/v2/products/?_page=2&category=3&format=json&material=2%2C3",
  "previous": null,
  ▼ "results": [
    ▼ {
      "id": 3536,
      "brandName": "Fossil",
      "productName": "Ohrstecker HEART silver",
      ▼ "productImages": [
        ▼ {
          "id": 3535,
          "description": null,
          "priority": 1,
          "imageUrl": "https://staging.cdn.prettique.de/product_images/52xrz6wvtlgo7e0f.jpg"
        }
      ],
      "productLink": "http://td.oo34.net/cl/?aaaid=VlpQAYKIMyDSTtH0&ein=52xrz6wvtlgo7e0f&paid=hwe48000icfo4rnf",
      "price": "38,95 €",
      "oldPrice": null,
      "foundAt": "Zalando",
      ▼ "categories": [
        ▼ {
          "id": 1,
          "name": "Ohrringe",
          ▼ "children": [
            ▼ {
              "id": 3,
              "name": "Ohrstecker"
            }
          ]
        }
      ],
      "priority": 1,
      "onWishlist": false
    },
    ,
  ]
}
```

GraphQL

- Strongly typed
- Only one URL for all actions
- Data are modeled as a graph
- Server describes data & actions
- Client asks only for data he wants
- Self-discovery of API is build-in
- Response shape is 1:1 with request shape

localhost:3001/graphql?query=%7B%0A%20%20%0A%7D

GraphQL

Prettify

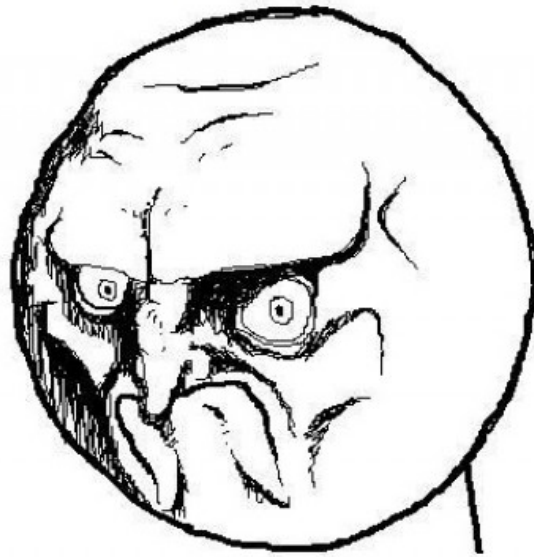
< Docs

1 {
2 |
3 }

QUERY VARIABLES

**I want real-live
communication!**

**Ask for data every
second.**



NO.

Ok, sometimes it can be usefull...
Ask your mentor / senior :)

Pushing

- Client registers on server for data
- Server is responsible for pushing the data to client
- Connection is long lived

GraphQL - subscription

- Part of the GraphQL specification
- Builds on top of existing GraphQL API
- Client subscribes to data-mutations done on server
- Server sends updated data every time they are changed

WebSocket

- Client opens connection to server
- Messages can then be sent both ways

Q&A

Thanks for attention!

<https://github.com/CorwinCZ/Client-server-communication>