SQL + REST == GraphQL

Prof. Dr. Alexander Paar

Duale Hochschule Schleswig-Holstein

Aufgabe: Projektmanagementsoftware

Die Projektmanagementsoftware soll Projekte und deren Auftraggeber verwalten.

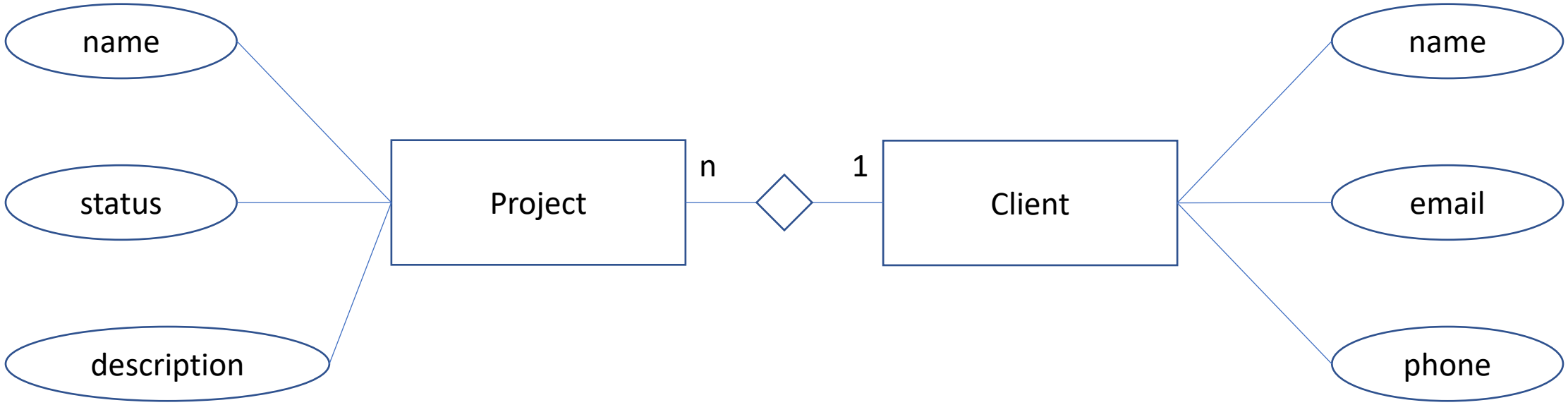
Projekte haben einen Namen, einen Status und eine Beschreibung.

Ein Projekt hat genau einen Auftraggeber.

Ein Auftraggeber hat einen Namen, eine Email-Adresse und eine Telefonnummer.

Ein Auftraggeber kann beliebig viele Projekte beauftragen.

Entity-Relationship-Modell



Relationales Modell

project

id	name	status	description	clientId
1	eCommerce Website	In Progress	Lorem ipsum…	1
2	Dating App	In Progress	Lorem ipsum…	2
3	SEO Project	In Progress	Lorem ipsum…	3
4	Design Prototype	Done	Lorem ipsum…	4
5	Auction Website	In Progress	Lorem ipsum…	5

client

id	name	email	phone
1	Tony Stark	ironman@gmail.com	343-567-4333
2	Natasha Romanova	blackwidow@gmail.com	223-567-3322
3	Thor Odinson	thor@gmail.com	324-331-4333
4	Steve Rogers	steve@gmail.com	344-562-6787
5	Bruce Banner	bruce@gmail.com	321-468-8887

SQL

```sql
create table client (
  id int not null,
  name varchar(100) not null,
  email varchar(100) not null,
  phone varchar(100) not null,
  primary key (id)
);
```

```sql
create table project (
  id int not null,
  name varchar(100) not null,
  status varchar(100) not null,
  description varchar(1000) not null,
  clientId int not null,
  primary key (id),
  foreign key (clientId) references client(id)
);
```

```sql
insert into client (id, name, email, phone)
values (
    1,
    "Tony Stark",
    "ironman@gmail.com",
    "343-567-4333"
  );
```

```sql
insert into project (id, name, status, description, clientId)
values (
    1,
    "eCommerce Website",
    "In Progress",
    "Lorem ipsum ...",
    1
  );
```

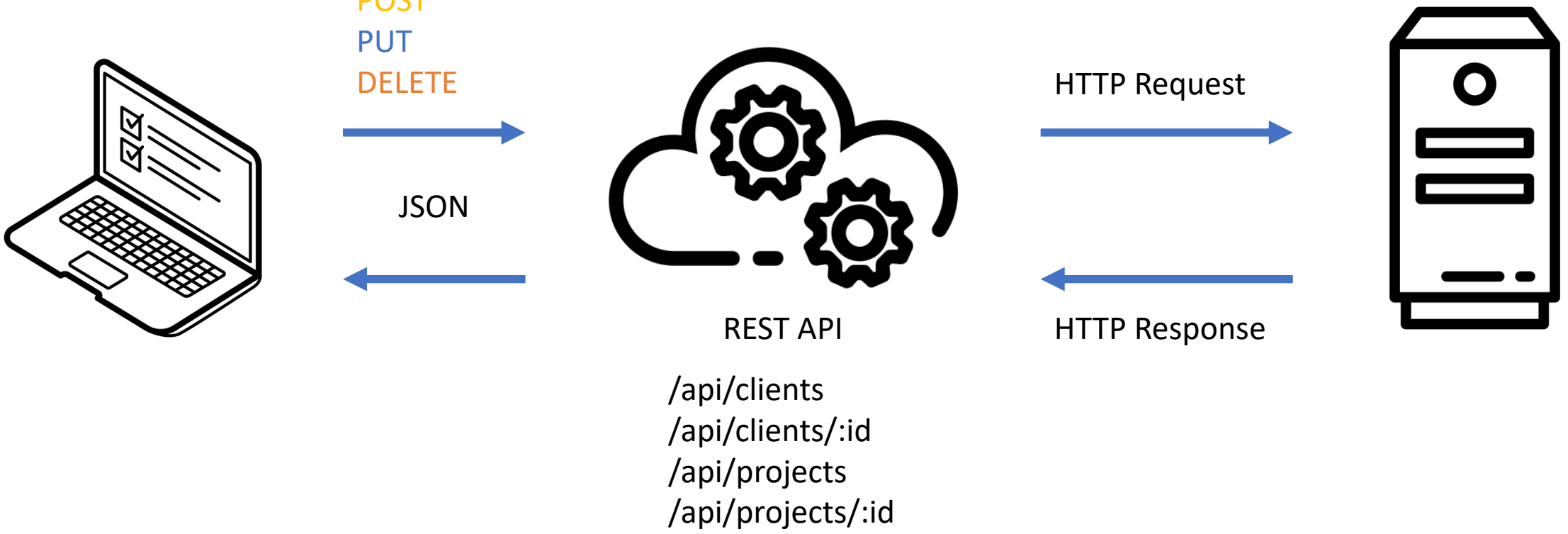
SQL

Was ist der Status und wie heißt der Auftraggeber von Projekt #1?

```
select p.name,
  p.status,
  client.name
from (
    select *
    from project
    where id = 1
  ) as p
  join client on p.clientId = client.id;
```

project.name	status	client.name
eCommerce Website	In Progress	Tony Stark

Eine webbasierte Projektmanagementsoftware



Was ist HTTP

Hypertext Transfer Protocol

Zustandsloses Protokoll zur Übertragung von Daten auf der Anwendungsschicht

HTTP Requests & Responses

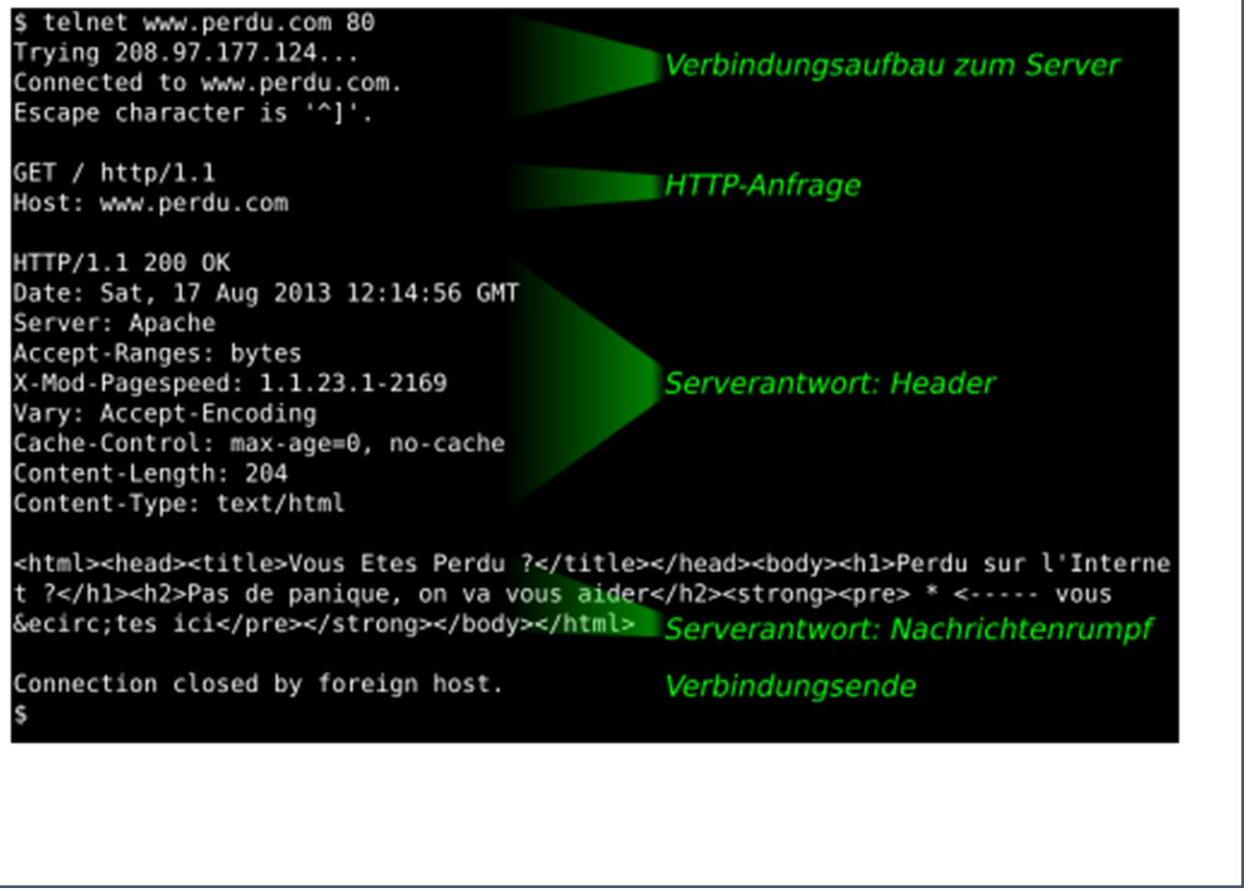
- Ein Server erhält Anfragen und sendet Antworten mit Statuscode zurück

HTTP Header

- Menge von Schlüssel-Werte-Paaren für zum Beispiel den „Content-Type“ oder ein Authentifizierungstoken

HTTP Body

- Daten die gesendet oder empfangen werden



HTTP Requests

GET

- Fordert die angegebene Ressource vom Server an

POST

- Fügt eine neue (Sub-)Ressource unterhalb der angegebenen Ressource ein

PUT

- Die angegebene Ressource wird angelegt. Wenn die Ressource bereits existiert, wird sie geändert.

DELETE

- Löscht die angegebene Ressource.

```
GET     http://example.com/api/users     // Get all users
GET     http://example.com/api/users/1   // Get single user 1
POST    http://example.com/api/users     // Add user
PUT     http://example.com/api/users/1   // Update user 1
DELETE  http://example.com/api/users/1   // Delete user 1
```

HTTP-Statuscodes

Ein HTTP-Statuscode wird von einem Server auf jede HTTP-Anfrage als Antwort geliefert

- Mitteilung, ob die Anfrage erfolgreich bearbeitet wurde

Die erste Ziffer eines Statuscodes stellt die Statusklasse dar

- Definiert in einigen RFCs
- Manchmal auch proprietäre Codes für Status- und Fehlermeldungen

https://de.wikipedia.org/wiki/HTTP-Statuscode

1xx – Informationen
2xx – Erfolgreiche Operation
- 200 OK
- 201 Created
- 204 No Content

3xx – Umleitung
- Not Modified

4xx – Client-Fehler
- 400 Bad Request
- 401 Unauthorized
- 404 Not Found

5xx – Server-Fehler
- 500 Internal Server Error
- 503 Service Unavailable

9xx – Proprietäre Fehler

REST API

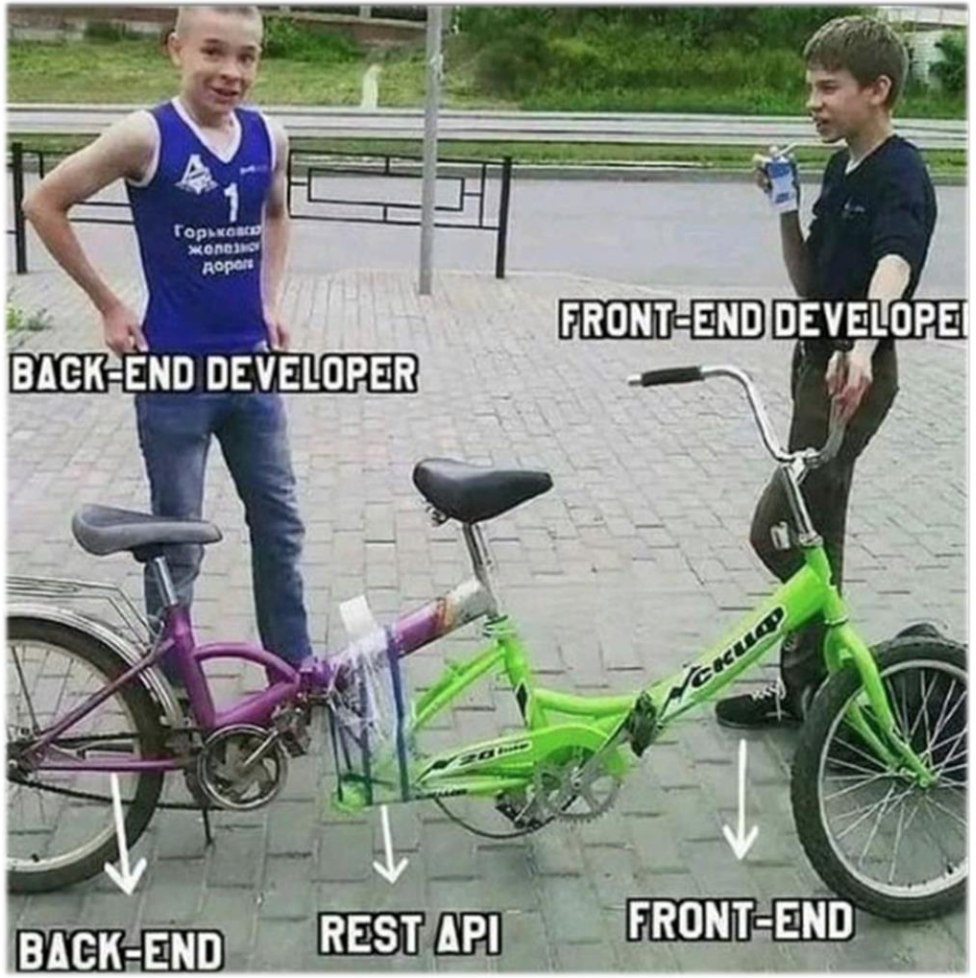
Was ist der Status und wie heißt der Auftraggeber von Projekt #1?

```
GET /api/projects/64271194e81c01c3ec14325d

{
    "_id": "64271194e81c01c3ec14325d",
    "client": "64271193e81c01c3ec14324c",
    "name": "eCommerce Website",
    "status": "In Progress",
    "description": "Lorem ipsum ...",
    "__v": 0
}
```

```
GET /api/clients/64271193e81c01c3ec14324c

{
    "_id": "64271193e81c01c3ec14324c",
    "name": "Tony Weak",
    "email": "ironman@gmail.com",
    "phone": "343-567-4333",
    "__v": 0
}
```

Problem: Overfetching & Underfetching & unnötige HTTP Requests

REST API…


BACK-END DEVELOPER
FRONT-END DEVELOPE
BACK-END
REST API
FRONT-END

GraphQL

Datenabfrage- und Manipulationssprache

Entwickelt 2012 von Facebook, veröffentlicht 2015, open sourced 2018

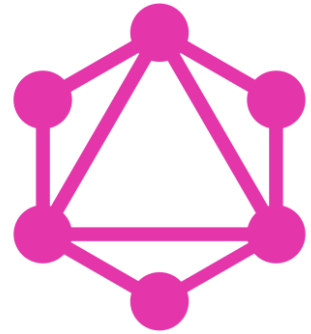
Zustandslose, parametrierbare Abfragen

GraphQL.js – JavaScript Referenzimplementierung (`graphql`)

- Typesystem, Abfragesprache, Ausführungssemantik, Validierung

graphql-http – GraphQL over HTTP Server (`graphql-http`)

GraphiQL – GraphQL IDE (`graphiql`)

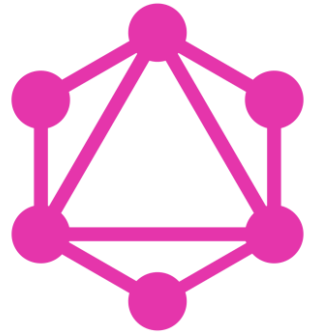


GraphQL

Was ist der Status und wie heißt der Auftraggeber von Projekt #1?

```
{
  project(id: "64271194e81c01c3ec14325d") {
    name
    status
    client {
      name
    }
  }
}
```

```
{
  "data": {
    "project": {
      "name": "eCommerce Website",
      "status": "In Progress",
      "client": {
        "name": "Tony Stark"
      }
    }
  }
}
```



Query-driven Schema Design

Ein GraphQL-Schema sollte entsprechend den Bedürfnissen (d.h. den Anfragen) der Klienten entworfen werden

Design your schema based on how data is *used*, not based on how it's *stored*

```
query EventList {
  upcomingEvents {
    name
    date
    location {
      name
      weather {
        temperature
        description
      }
    }
  }
}
```

Query

Schema

```
type Query {
  upcomingEvents: [Event!]!
}

type Event {
  name: String!
  date: String!
  location: Location
}

type Location {
  name: String!
  weather: WeatherInfo
}

type WeatherInfo {
  temperature: Float
  description: String
}
```

https://www.apollographql.com/docs/apollo-server/schema/schema/#query-driven-schema-design

Tools

Docker
SQL
MySQL
MySQL Workbench
JSON
MongoDB
MongoDB Compass
Visual Studio Code + Extensions
HTTP
Node.js
Express.js
Postman
React
GraphQL
GraphiQL
Apollo Client

SQL + REST == GraphQL

Die GraphQL-Spezifikation erwähnt kein HTTP!

POST um eine Ressource zu lesen?

Analytics, Monitoring, Logging, Caching?

Wie wäre es anstatt SQL + REST mit…

GraphQL + REST ?

DH DUALE
SH HOCHSCHULE SH