REST Intro

- **Re** presentational
- **S** tate
- **T** ransfer

2000, Dr. Roy Fielding - PhD Thesis, WWW-Consortium

Nutzen wir die Kern-Prinzipien der darunterliegenden Technologien → **HTTP**, **HTML**

HTTP Basics / History

Http

- **H** ypter **t** ext
- **T** ransfer
- **P** rotocol

Http Methoden / Operationen (aka Verben)

- Get
- Post … Create
- Put … Replace
- Delete
- Patch … Update
- Head
- Connect
- Trace
- Option

Html

späten '80er / frühe '90er, Sir Tim Berners Lee, CERN

- Header 1-6
- Paragraph
- Quote
- Link
- Basic text format (bold, italic, underline)

Alternativen zu REST

- RPC (Java RMI)
- Corba
- SOAP

REST arbeitet mit Ressourcen, welche über URI and URL identifiziert werden können.

Wie kommt man von dem da oben zu REST → Richardson Maturity Model

RESTful vs REST (Maturity Level 2 vs 3)

APIGee - API Design Principles

Common RESTful ressource routes

Operation	/api/albums (0..n)	/api/albums/{key} (0..1)
GET	y	y
POST	y	n/a
PUT	? (n)	y *
PATCH	? (y)	y
DELETE	? (n)	y *

- idempotent

HTTP Status Codes

- 100er … informell
- 200er … OKish
- 300er … Weiterleitungen … Suchmaschinen SEO
- 400er … Client Error … 418 (April Fools Status)
- 500er … Server Error
- 900er … Custom

Wichtige HTTP Status Codes:

200, 201, 204, 400, 403(! → 404), 404, 503

Complexity bei der Abfrage → QueryString

QueryString: URL?owner=ACH&updatedWithin=2w

- Alle Alben, die Andreas Chwatal gehören /api/albums?owner=ACH

- Alle Alben, die in den letzten ?? geändert wurden /api/albums?updatedWithin=2w

Nesting von Collections

/api/albums/{key}/photos?startPos=1&endPos=17

Operations via REST abbilden

zB Wechsel von Währungen

/api/currency/{key} … /api/currency/USD

- /api/currency/convert?src=USD&target=EUR … default amount=1 timestamp=now
- /api/currency/convert?src=USD&target=EUR&amount=100
- /api/currency/convert?src=USD&target=EUR&amount=100&timestamp=20231130214500