

TechLab 12.09.2019

API-Design mit OpenAPI

Manuel Ottlik
Carl Volhard

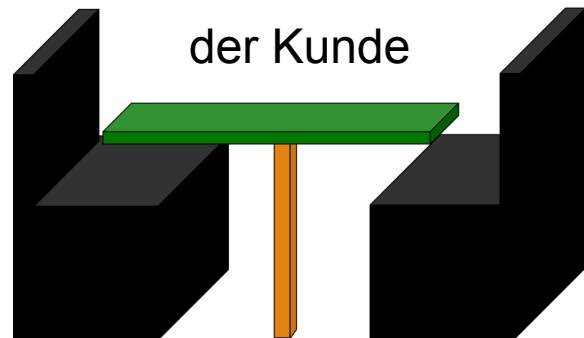
Agenda

- Vorstellungsrunde
- Was ist eine API?
- Abgrenzung REST vs. SOAP
- Rest-Paradigma
- Einführung OpenAPI
- Swagger-UI
- Use Case vorstellen
- Hands-on

Vorstellungsrunde

Was ist eine API?

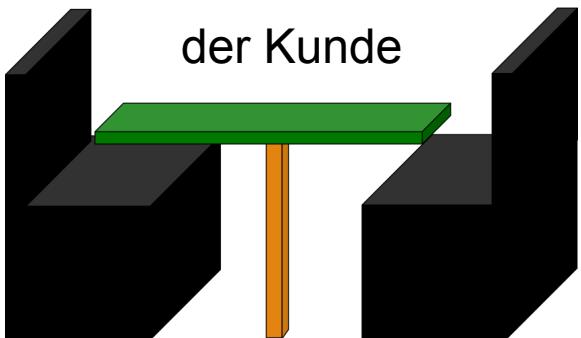
Herzlich Willkommen im RESTaurant!



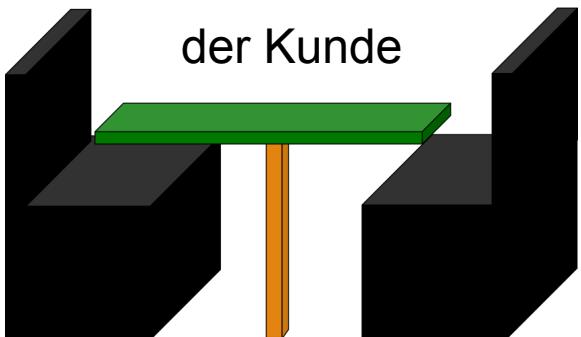
die Küche



Herzlich Willkommen im RESTaurant!



Herzlich Willkommen im RESTaurant!



der Kunde



die Speisekarte

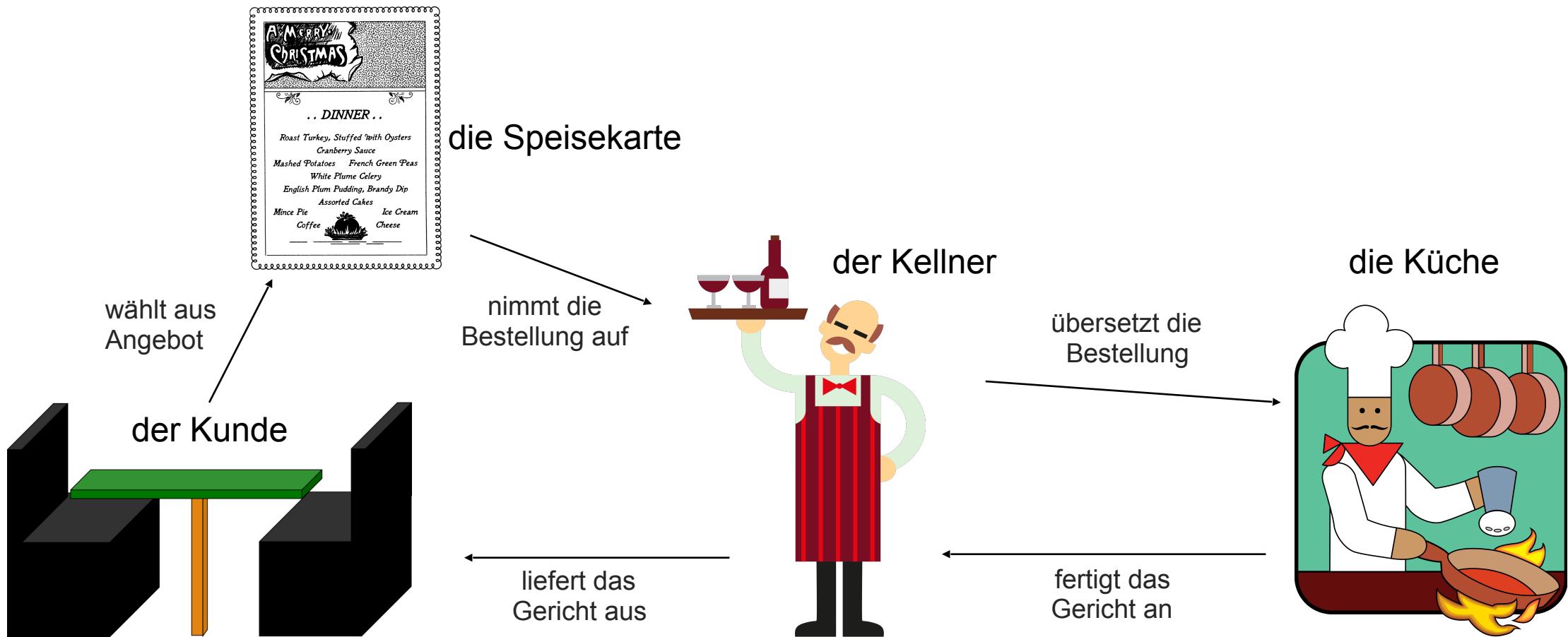


der Kellner

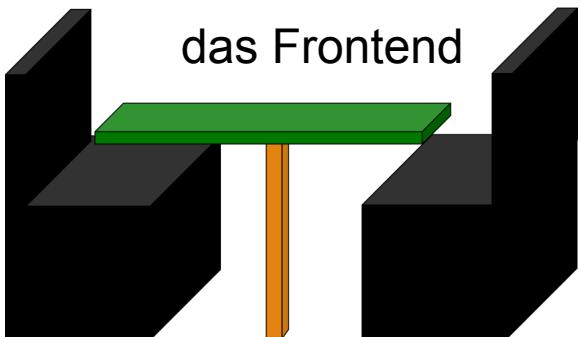


die Küche

Herzlich Willkommen im RESTaurant!



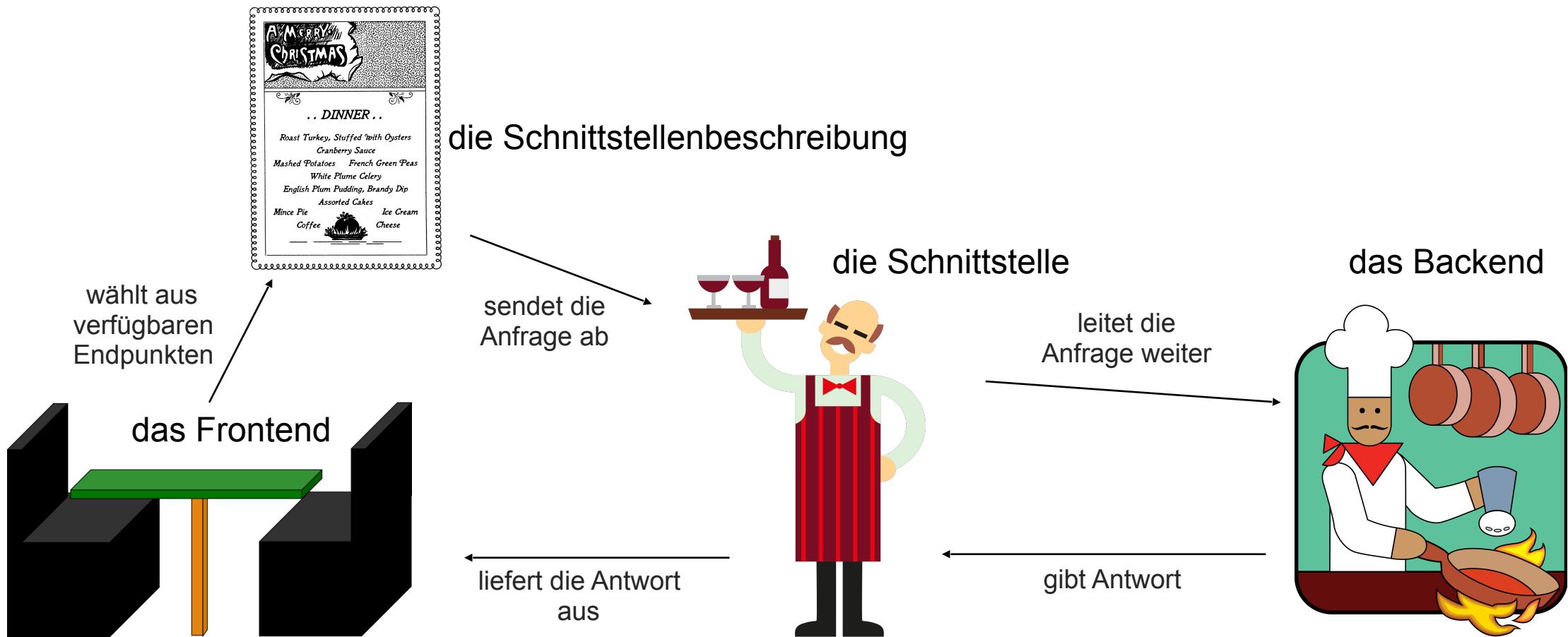
Herzlich Willkommen im RESTaurant!



die Schnittstellenbeschreibung



Herzlich Willkommen im RESTaurant!



SOAP & XML vs. REST & JSON

Strukturierung

SOAP



aktionsorientiert

REST



ressourcenorientiert

Basis

SOAP



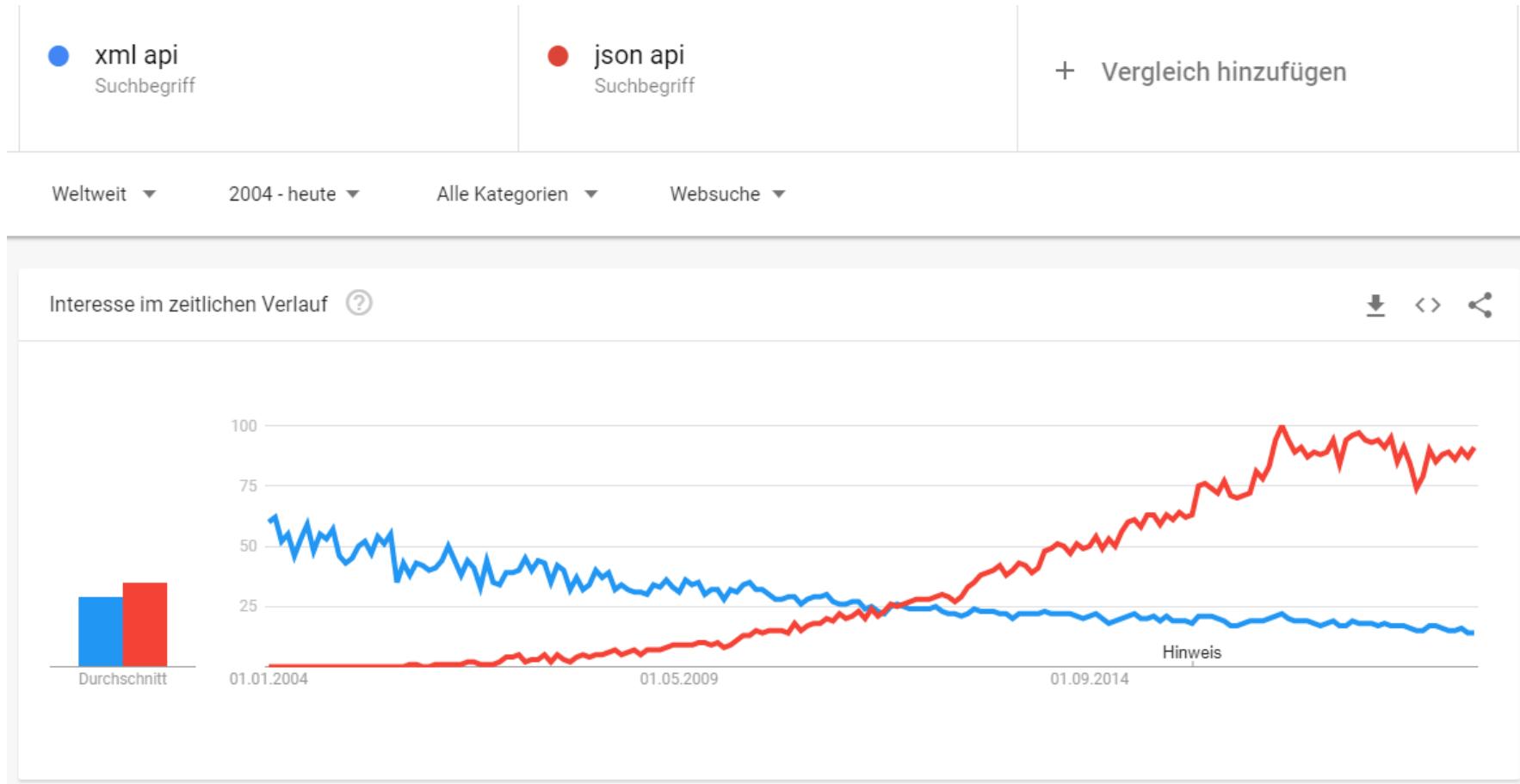
Remote Procedure Calls

REST



Hypertext Transfer Protocol

Beliebtheit



Informationsdichte

XML

```
<empinfo>
  <employees>
    <employee>
      <name>James Kirk</name>
      <age>40</age>
    </employee>
    <employee>
      <name>Jean-Luc Picard</name>
      <age>45</age>
    </employee>
    <employee>
      <name>Wesley Crusher</name>
      <age>27</age>
    </employee>
  </employees>
</empinfo>
```

JSON

```
{ "empinfo" :
  {
    "employees" : [
      {
        "name" : "James Kirk",
        "age" : 40,
      },
      {
        "name" : "Jean-Luc Picard",
        "age" : 45,
      },
      {
        "name" : "Wesley Crusher",
        "age" : 27,
      }
    ]
  }
}
```

Was ist das REST-Paradigma?

REpresentational S_tate T_{ransfer}

- ursprünglich behandelt in der Dissertation von Roy Fielding
- kein Regelwerk, sondern Sammlung von Best Practices
- zustandslos
- über gibt Abbild eines Zustandes (State Transfer)
- kann mehrere Repräsentationen einer Ressource anbieten

Ressourcenorientierung

GET

POST

PUT /offices/{id}

PATCH

DELETE

Verschachtelung

GET

POST

PUT

/offices/{id}/desks/{id}

PATCH

DELETE

Operationen auf eine Ressource

Aktion	HTTP-Methode	URI
alle Büros auslesen	GET	/offices
neues Büro anlegen	POST	/offices
Büro mit der ID {x} auslesen	GET	/offices/{id}
Büro mit der ID {x} komplett aktualisieren	PUT	/offices/{id}
Büro mit der ID {x} teilweise aktualisieren	PATCH	/offices/{id}
Büro mit der ID {x} löschen	DELETE	/offices/{id}
alle Schreibtische, die in dem Büro mit der ID {x} stehen, auslesen	GET	/offices/{id}/desks/
einen neuen Schreibtisch anlegen, der in dem Büro mit der ID {x} steht	POST	/offices/{id}/desks/
Schreibtisch mit der ID {y} auslesen, der in dem Büro mit der ID {x} steht	GET	/offices/{id}/desks/{id}
...		

Antworten mit HTTP-Statuscodes

GET	/offices	200	OK
POST	/offices	201	Created
DELETE	/offices/1	202	No Content

Antworten mit HTTP-Statuscodes

GET	/offices	400 Bad Request
GET	/offices	401 Unauthorized
GET	/offices/2	403 Forbidden
GET	/offices/7	404 Not Found
POST	/offices	422 Unprocessable Entity

Antworten mit HTTP-Statuscodes

3xx Umleitungen

5xx Server-Fehler

Anpassung der Antwort

Content-Type: application/json
Accept: application/json
Accept: application/xml
Accept: application/ical
Accept: application/csv
Accept-version: v10
Authorization: Bearer: apsofijawßef9ijavsdvoinam49
Cache-Control: max-age=300

OpenAPI

API-Beschreibungen

- Formale Beschreibung der API
- Nur Beschreibung, keine Implementierung
- Beschreibungsformate
 - API Blueprint
 - RAML
 - OpenAPI



<https://nordicapis.com/top-specification-formats-for-rest-apis/>

OpenAPI Entstehung

- Version 1.0.0 2011 (Swagger)
- Apache 2.0 Open-Source-Lizenz
- Seit 2014 populäre als RAML und API Blueprint
- 2015 von SmartBear an die Linux Foundation
- 2016 in OpenAPI-Specification (OAS)
- 2017 Version 3.0.0
- Aktuell Version 3.0.2

Exkurs YAML

- OAS wird in YAML/JSON geschrieben
- „YAML Ain't Markup Language“
- 2001 spezifiziert
- Zum Austausch/Speichern von beliebigen Datenstrukturen
 - Skalare
 - Listen (Arrays)
 - Objekte (Dictionaries/Maps)
- Superset von JSON
- Whitespace-sensitiv
- Menschenlesbar(er als XML)

Exkurs YAML

```
# Kommentar
id: 731
customer: 17
items:
- quantity: 5
  description: Oolong
  price: 5.98
- quantity: 2
  description: Assam
  price: 2.95
shippingAddress:
  street: Friedrichswall 10
  zip: 30159
  city: Hannover
```

Skalare Werte

Liste von Objekten

Einzelnes Objekt

OAS Specification

openapi: 3.0.2

info:

title: Raumbuchungs-API

version: 0.1.0

description: API für das Buchen ...

servers:

- **url:** https://test.de/raumbuchungsapi
description: Testumgebung
- **url:** https://dev.de/raumbuchungsapi
description: Entwicklungsumgebung

tags:

- **name:** Besprechungsräume
- **name:** Raumbuchungen

security:

- **BasicAuth:** [read, write]
- **bearerAuth:** []

components:

[...]

paths:

[...]

externalDocs:

description: zusätzliche Doku zur API
url: https://my.doku.domain/fachliches

OAS Specification - Components

```
#openapi: 3.0.2
components:
  schemas:
    RoomWrite:
      type: object
      description: ... schreibende Zugriffe
      required:
        - name
        - capacity
      properties:
        name:
          type: string
          description: Raumbezeichnung
          example: 00_00W50
        capacity:
          type: integer
          description: Anzahl Sitzplätze ...
          example: 25
```

```
#schemas:
  Room:
    type: object
    description: ... lesende Zugriffe
    required:
      - room_id
    allOf:
      - type: object
        properties:
          room_id:
            type: string
            description: UUID des Raumes
            example: 6f410920-ea6b-4fbe-...
      - $ref: '#/components/schemas/RoomWrite'
```

OAS Specification - Components

```
#components:  
  parameters:  
    RoomId:  
      name: room_id  
      in: path  
      required: true  
      description: UUID des Besprechungsraumes.  
      schema:  
        type: string  
        example: 5fcec5cd-a4f6-42f3-b9c0-...
```

OAS Specification - Paths

```
#openapi: 3.0.2
paths:
  /rooms:
    description: Ressource für Bespre...
    get:
      summary: Liefert Besprechungsräume
      description: Liefert alle ...
      tags:
        - Besprechungsräume
      operationId: getRooms
      parameters:
        - name: capacity
          description: Parameter um ...
          in: query
          schema:
            type: integer
```

```
#get:
  responses:
    '200':
      description: Erfolgreicher Zugriff.
      content:
        application/json:
          schema:
            type: array
            items:
              $ref: '#/components/schemas/Room'
      example:
        - room_id: 6f410920-ea6b-4fb...
          name: 00_00W50
          capacity: 25
        - room_id: fb904c1b-bc00-4ce...
          name: 00_00W60
          capacity: 42
```

OAS Specification - Paths

```
#responses:  
  default:  
    description: Fehlerhafter Zugriff  
    content:  
      application/json:  
        schema:  
          $ref: '#/components/schemas/Error'
```

OAS Specification - Paths

```
#/rooms:  
  post:  
    summary: Anlage eines Besprechungraumes.  
    description: Legt einen neuen Besprechungsraum ...  
    tags:  
      - Besprechungsräume  
    operationId: addRoom  
    requestBody:  
      required: true  
      content:  
        application/json:  
          schema:  
            $ref: '#/components/schemas/RoomWrite'  
    responses:  
      '200':  
        description: Neuer Besprechungsraum wurde ...  
        content:  
          application/json:  
            schema:  
              $ref: '#/components/schemas/Room'
```

OAS Specification - Paths

```
#paths:  
  /rooms/{room_id}:  
    description: Resource um auf einen einzelnen ...  
    parameters:  
      - $ref: '#/components/parameters/RoomId'  
    get:  
      # ...  
    put:  
      # ...
```

Swagger-UI

Swagger-UI

The screenshot illustrates the Swagger-UI interface for the Raumbuchungs-API. The top navigation bar includes the Swagger logo, a search bar, and a green 'Explore' button. The main content area features the API title 'Raumbuchungs-API' (version 0.1.0, OAS3), a brief description, and a detailed info block.

openapi: 3.0.2

info:
title: Raumbuchungs-API
version: 0.1.0
description: API für das Buchen ...

Servers
https://gravitee-01.test.k8s.i-lab.online/raumbuchungsapi - Testumgebung

tags:
- ...
- ...

Besprechungsräume >
Raumbuchungen >

Components:
schemas:

Swagger-UI

Besprechungsräume

GET

/rooms Liefert Besprechungsräume.

Liefert alle Besprechungsräume zurück. Wurden Parameter übergeben, werden nur die Besprechungsräume zurückgeliefert auf die **alle** Kriterien zu treffen. Wenn es keine Räume gibt, auf die die Filterkriterien zu treffen, wird ein 200er mit leerer Liste zurückgeliefert.

Parameters

Try it out

Name	Description
------	-------------

capacity integer (query)	Parameter um nach Besprechungsräumen zu suchen, die mindestens Platz für die angegebene Anzahl Personen hat.
--------------------------------	--

Responses

Code	Description	Links
------	-------------	-------

200	<p>Erfolgreicher Zugriff.</p>	No links
-----	-------------------------------	----------

application/json

Controls Accept header.

Use Case

Use Case

- Raumbuchungs-API erweitern
- Ressourcen für die eigentliche Buchung
- Buchungsobjekt
 - ID
 - Raum-ID
 - Start
 - Dauer
 - Email-Adresse des Reservieren (optional)

Links

- GitHub-Projekt: github.com/nordlb > techtalks > 01-api-design
- OpenAPI
 - Online Editor: editor.swagger.io
 - Specification: swagger.io/specification/
 - Documentation: swagger.io/docs/specification/about/
- REST
 - Dissertation Roy Fielding: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf
 - HTTP RCFs: [RFC7231](https://tools.ietf.org/html/rfc7231), [RFC7230](https://tools.ietf.org/html/rfc7230), [RFC6585](https://tools.ietf.org/html/rfc6585)