



REST API



REpresentational **S**tate **T**ransfer **A**pplication **P**rogramming **I**nterface



Inhaltsverzeichnis:

- Allgemeines zu API
- Was ist REST?
- Eigenschaften von REST
- Umsetzung von Rest mit HTTP/S
- Vor- und Nachteile von REST
- Quellenverzeichnis

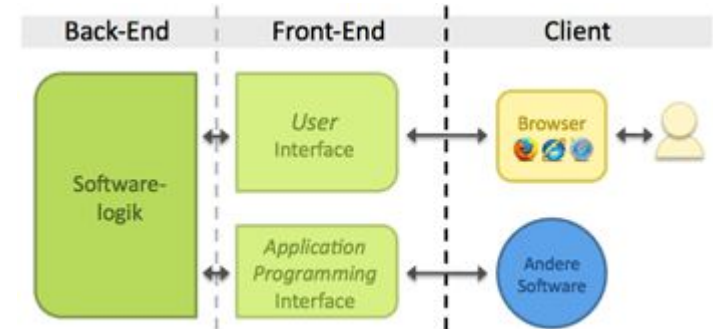


Allgemeines zu API

- Schnittstelle, die anderen Programmen zur Verfügung gestellt wird
- Begriff API hat vor allem Relevanz bei Verwendung durch Webdiensten
- dient zum Austausch und der Weiterverarbeitung von Daten und Inhalten
- maschinelle Äquivalent zum User-Interface

Allgemeines zu API

- Austausch in gut weiterzuverarbeitenden und reduzierten Form
- maschinenlesbar
- Dateiformat zur Rückgabe von Daten: meistens XML oder JSON





Beispiel

- Über YouTube-API kann nach Videos mit gewünschten Parametern gesucht werden.
- Die API schickt Antwort in Form einer JSON-Datei zurück.
- Diese kann ausgewertet und für eigene Zwecke verwendet werden.
- Anfrage: https://www.googleapis.com/youtube/v3/videos?id=9bZkp7q19f0&part=contentDetails&key=API_KEY



Beispiel Rückgabe

```
{
  "kind": "youtube#videoListResponse",
  "etag": "\"XlbeM5oNbUofJuiuGi6IkumnZR8/ny1S4th-ku477VARrY_U4tIqcTw\"'",
  "items": [
    {
      "id": "9bZkp7q19f0",
      "kind": "youtube#video",
      "etag": "\"XlbeM5oNbUofJuiuGi6IkumnZR8/HN8ILnw-DBXyCcTsc7JG0z51BGg\"'",
      "contentDetails": {
        "duration": "PT4M13S",
        "dimension": "2d",
        "definition": "hd",
        "caption": "false",
        "licensedContent": true,
        "regionRestriction": {
          "blocked": [ "DE" ]
        }
      }
    }
  ]
}
```



Was ist REST ?

- REST ist ein Kommunikationsprinzip für verteilte Systeme
 - ◆ also kein konkretes Protokoll
- dient zum Abrufen von Informationen von einem Server
- Alternative für:
 - ◆ SOAP (*Simple Object Access Protocol*)
 - ◆ WSDL (Web Services Description Language)



- Webseiten mit einem statischen Format, somit weite Teile des WWW sind REST-konform
- Schwerpunkt liegt bei der Kommunikation zwischen zwei Maschinen
- Aufruf erfolgt über das HTTP/S-Protokoll



Eigenschaften von REST

- Client-Server-Modell
- Zustandslosigkeit
 - ◆ jede Anfrage enthält alle nötigen Informationen (keine Cookies oder Sessions)
- Cacheable



Eigenschaften von REST

- *Einheitliche Schnittstelle*: Anfragen folgen dem CRUD-Prinzip (Create, Read, Update, Delete)
- Universelle Rückgabe (HTML, JSON, XML)
 - ◆ entspricht meist nicht der internen Datenstruktur
- *Adressierbarkeit*: Jede Ressource muss über einen eindeutigen ***Unique Resource Identifier*** (URI)



Umsetzung von REST mit HTTP/S

→ Aufruf einer URI über den Port 80 bzw. 443:

[HTTP-Befehl] https://api.beispiel.de/rest/resources/

<u>CRUD</u>	Create	Read	Update	Delete
<u>HTTP-Befehl</u>	POST	GET	PUT	DELETE



Umsetzung von REST mit HTTP/S

- **GET** dient dazu, *lesend* auf Ressourcen zuzugreifen.
- Mit einem **POST**-Request können *neue Ressourcen erstellt* werden (noch unbekannte URI)
- Ein **PUT**-Request wird verwendet, um *Ressourcen zu erstellen oder zu bearbeiten* (bekannte URI)
- Mit einem **DELETE**-Request können Ressourcen gelöscht werden.



Beispiel POST

POST http://MyService/Person/

Host: MyService

Content-Type: text/xml; charset=utf-8

Content-Length: 123

<?xml version="1.0" encoding="utf-8"?>

<Person>

<ID>1</ID>

<Name>tobias Boge</Name>

<Email>tboge@gmail.com</Email>

<Country>World</Country>

</Person>



REST Hypermedia

```
<order href='http://example.com/customers/1234' >  
  <amount>23</amount>  
  <product href='http://example.com/products/4554' />  
  <customer href='http://example.com/customers/1234' />  
  <link rel='cancel' href='./cancellations' />  
</order>
```



HTTP Statuscodes

200	OK
201	CREATED (PUT oder POST)
400	BAD REQUEST
401	UNAUTHORIZED
404	NOT FOUND
405	METHOD NOT ALLOWED
409	CONFLICT (z.B. PUT zweimal)
500	INTERNAL SERVER ERROR



Vorteile von REST

- Sprachen und Plattform unabhängig
- Einfacher zu entwickeln als Alternativen
- Bessere Performance bzw. weniger benötigte Bandbreite
- Client-seitige Entwicklung wesentlich einfacher da HTTP
- Für Menschen lesbare Ergebnisse



Nachteile von REST

- Keine Standards
- Server-seitige Entwicklung aufwendiger
- Wird hauptsächlich zum Abfragen von Daten (GET) verwendet
- HTTPS relativ unsicher



Vielen Dank für Ihre Aufmerksamkeit



Bestehen noch Fragen ?



Quellenverzeichnis:

Thomas Bayer, <http://www.oio.de/public/xml/rest-webservices.htm>, 28.09.2015

Dr. M. Elkstein, <http://rest.elkstein.org/>, 02.10.2015

Roy Thomas Fielding, https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm, 02.10.2015

Julia Duong, <http://www.xfront.com/REST-Web-Services.html>, 02.10.2015

http://www.dpunkt.de/leseproben/3574/3_Einfuehrung%20in%20REST.pdf 05.10.2015