

# RESTful API mit node.js

AUTHOR

11. Februar 2018

## **Zusammenfassung**

Dieses Dokument beschreibt die Theorie hinter REST<sup>1</sup> und die Implementierung einer RESTful API<sup>2</sup> mithilfe von node.js

---

<sup>1</sup>Representational State Transfer

<sup>2</sup>Application Programming Interface

# Inhaltsverzeichnis

<b>1</b>	<b>Allgemeines über REST</b>	<b>3</b>
1.1	Ressourcen . . . . .	3
1.2	HTTP <sup>3</sup> -Methoden . . . . .	3
	<b>Abkürzungen</b>	<b>5</b>
	<b>Literaturverzeichnis</b>	<b>6</b>

---

<sup>3</sup>Hypertext Transfer Protocol

# 1 Allgemeines über REST

REST ist ein Programmierparadigma und wurde von Roy Fielding in seiner Dissertation[2] spezifiziert.

Die wichtigsten Eigenschaften[1, p. 2] sind:

- Alles ist eine Ressource und jede ist mit einer eindeutigen Adresse identifizierbar (URI<sup>4</sup>)
- Verwendung von Standard-HTTP Methoden<sup>5</sup>
- Ressourcen können mehrerer Repräsentationen besitzen
- Zustandslose Kommunikation
- HATEOAS<sup>6</sup>

## 1.1 Ressourcen

Die Ressourcen sind unter einer eindeutigen Adresse (URI) erreichbar und können unter verschiedenen Repräsentationen dargestellt werden:

Zum Beispiel kann ein Ressource folgende URI haben "http://ipadress:port/path/Ressource"

## 1.2 HTTP-Methoden

Die Verwendung von HTTP-Methoden ist keine stricte Voraussetzung für REST. In der Praxis wird jedoch hauptsächlich HTTP verwendet für REST. Deshalb wird hier auch nur auf die Implementierung mit HTTP-Methoden eingegangen.

Es sind folgende Operationen laut RFC2616[3] im HTTP-Protokoll möglich:

- GET
- POST
- PUT
- DELETE
- HEAD
- OPTIONS
- TRACE
- CONNECT

Für REST werden die folgenden 4 CRUD<sup>7</sup>-Operationen verwendet.

Create	POST
Read	GET
Update	PUT
Delete	DELETE

---

<sup>4</sup>Unique Adress Identifier

<sup>5</sup>Keine strenge Voraussetzung für REST, in der Praxis wird aber hauptsächlich HTTP verwendet

<sup>6</sup>Hypermedia as the Engine of Application State

<sup>7</sup>Create Read Update Delete

asdasd

## Abkürzungen

**REST** Representational State Transfer

**API** Application Programming Interface

**URI** Unique Address Identifier

**HTTP** Hypertext Transfer Protocol

**HATEOAS** Hypermedia as the Engine of Application State

**CRUD** Create Read Update Delete

## Literaturverzeichnis

- [1] Valentin Bojinov. *RESTful Web API Design with Node.js*. Packt Publishing, 2015. ISBN 978-1-78398-586-9.
- [2] Roy Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, 2000. <https://www.ics.uci.edu/fielding/pubs/dissertation/fielding.dissertation.2up.pdf>, visited 2018-02-09.
- [3] Roy T. Fielding, James Gettys, Jeffrey C. Mogul, Henrik Frystyk Nielsen, Larry Masinter, Paul J. Leach, and Tim Berners-Lee. Hypertext transfer protocol – http/1.1. RFC 2616, RFC Editor, June 1999. URL <http://www.rfc-editor.org/rfc/rfc2616.txt>. visited 2018-02-09.