



## HTTP & REST Begrifflichkeiten

REST an sich steht für Representational State Transfer. Es bezieht sich dabei auf die Repräsentation des Zustands einer Ressource, die mittels des Http Protokolls vom Server an den Client übertragen wird.

Dabei kann der Zustand zum Zeitpunkt der Client Anfrage auf verschiedenste Weisen zurückgegeben werden, z.B. als

- HTML
- JSON (üblichste in einer modernen Web API)
- XML
- Text
- Csv
- Uvm.

REST wird oft als eine Möglichkeit betrachtet CRUD APIs anzulegen.

CRUD steht dabei für Create, Read, Update und Delete. Und das ist natürlich auch nicht falsch.

Mit den CRUD Operationen kann man also für eine bestimmte Ressource ermöglichen, dass diese angelegt, gelesen und verändert wird. Also fundamentale Operationen auf dieser Ressource zugelassen werden.

## HTTP Verben/Methoden

Im Header eines HTTP Requests wird vor der URL angegeben, um welche Methode es sich handeln soll. Dies bestimmt dann üblicherweise, welche Operation am Server ausgeführt werden soll. Das ist hilfreich, da man so ein (ausgehend davon dass sich alle an die gängigen Konventionen halten) erwartbares Resultat erzielt.

## HTTP Methoden, CRUD und REST

In RESTful APIs nutzt man für diese CRUD Operationen dann üblicherweise die Kombination aus den folgenden Http Methoden, bzw. Verben.

HTTP VERB	CRUD OPERATION
POST	Create
GET	Read
PUT	Update
DELETE	Delete



## Idempotenz

Ein weiterer wichtiger Aspekt bei einer Restful API ist dabei die Idempotenz. Idempotenz bedeutet, dass eine am Server ausgeführte Aktion immer das gleiche Ergebnis zur Folge hat, unabhängig davon, wie oft diese ausgeführt wird.

Ändere ich z.B. den Namen einer Customer Ressource von Customer-A zu Customer-42, dann kann ich das 100te Male nacheinander aufrufen aber die Ressource wird nach jedem Aufruf den Namen Customer-42 als Zustand aufweisen.

## Idempotenz und REST

Im Sinne von REST verhalten sich die http Methoden wie folgt:

POST / Create	NICHT idempotent
GET / Read	IMMER idempotent
UPDATE/ Update	IMMER idempotent
DELETE / Delete	Idempotent nach dem ersten Aufruf