

# Fortgeschrittene Programmierung

Vorlesungsfolien Sommer 2018  
Thomas Richter nach Unterlagen v. Dietmar Jannach

# Geschichte des Internets

# Geschichte des Internets

---

## ▶ ARPAnet und Internet

- ▶ 1958 Gründung des Advanced Research Projects Agency (ARPA)
  - ▶ Untersteht dem US Verteidigungsministerium
- ▶ 1969 Geburtsjahr des Arpanet
  - ▶ Universitäten vernetzen sich
  - ▶ Das Netzwerk wird dezentral und redundant ausgebaut
- ▶ 1978 Einsatz von TCP/IP im Arpanet
  - ▶ Datenübermittlung über einzelne Datenpakete
  - ▶ Grundlage für das globale Internet
- ▶ 1988 Die ersten deutschen Internetprovider
  - ▶ „Eunet“ in Dortmund
  - ▶ „Xlink“ in Karlsruhe
- ▶ 1989 Arpanet wird abgeschaltet



# Geschichte des Internets

## Geschichte des WWW

---

### ▶ World Wide Web

- ▶ Erfinder: **Tim Berners-Lee**
  - ▶ Idee: Weltweiter Zugriff auf Forschungsergebnisse von Wissenschaftlern
- ▶ Besonderheiten des WWW:
  - ▶ **HTTP** = Hypertext Transfer Protocol
    - Kommunikationsprotokoll zwischen Web-Client und Web-Server
  - ▶ **URI** = Uniform Resource Identifier
    - Zur Adressierung beliebiger Dateien und Datenquellen im Web
  - ▶ **HTML** = Hypertext Markup Language
    - Auszeichnungssprache für Web-Dokumente zur Anzeige in Browsern
- ▶ Webseite des ersten Webservers: <http://info.cern.ch> (noch immer online)



# Geschichte des Internets

## Geschichte des WWW

---

### ▶ World Wide Web Consortium (W3C)



- ▶ Gründung: 1994 durch Tim Berners-Lee
- ▶ Zweck: Standardisierungsgremium für das WWW
- ▶ Offene Standards sollten verhindern, dass das Internet in kleinere Teilnetzwerke zerfällt
- ▶ Entwicklung technischer Web-Standards in zyklischen, festgelegten Prozessen mit Review-Verfahren, z.B. HTML 5, CSS 3, XML 1.0
- ▶ Aufgabenbereiche:

Vertrauen  
stärken

Semantisches  
Web

Interoperabilität

Attraktiveres  
Multimedia

Entwicklungsfähigkeit

Dezentralisierung

Universelle  
Zugangsmöglichkeiten

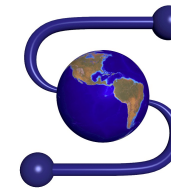
# Geschichte des Internets

## Geschichte des WWW

---

### ▶ Mosaic-Browser

- ▶ Erster Browser für grafische Benutzeroberflächen
- ▶ Später Umbenennung in Netscape
- ▶ Entwickler: Marc Andreessen
  - ▶ 90% Marktanteil des Netscape-Browsers von 1995 – 1996
  - ▶ Neues Berufsbild: „Webdesigner“



### ▶ Internet Explorer

- ▶ Mitte 90er steigt Microsoft in den Markt ein
- ▶ 1997 entsteht die 4er-Version des Internet Explorers
  - ▶ Erreicht in kurzer Zeit ähnliche Erfolge wie Netscape
  - ▶ Der sogenannte Browser-Krieg beginnt



# Geschichte des Internets

## Geschichte des WWW

---

### ▶ Mozilla

- ▶ 1998 wird der Quellcode von Netscape veröffentlicht
- ▶ Das Mozilla-Projekt wird gegründet
- ▶ Firefox wird auf Basis von Netscape entwickelt (Open-Source)
- ▶ Starke Verbreitung ab 2004 dank nützlicher Features



### ▶ Google

- ▶ 2008 erste Version des Chrome Browsers
- ▶ Auf Basis der Open-Source-Version Chromium
- ▶ Seit 2017 Marktführer in Deutschland



### ▶ Aktuelle Verbreitung in Deutschland (statista.com, März 2018)



37,6%



26,9%



13,2%



9,7%



3,3%

# Geschichte des Internets

## Geschichte des WWW

---

- ▶ Smartphones
  - ▶ 2007 erstes iPhone
  - ▶ 2008 erstes Android-Smartphone (HTC Dream)
- ▶ **Aktuell 52% Browser-Traffic von Mobil** (gs.statcounter.com, März 2018, mobile ist Marktführer seit Nov. 2016)

- ▶ **Neue Ansprüche an Websites & Dienste**

- ▶ Kleiner Bildschirm (Responsive Design)
- ▶ Begrenzte Bandbreite
- ▶ Geringere Rechenleistung
- ▶ Interaktion nur über Touch





# HTTP – Hypertext Transfer Protocol

# HTTP – Hypertext Transfer Protocol

---

- ▶ **Kommunikationsprotokoll**
  - ▶ Dient der Übertragung von Dateien zwischen Webclient und Webserver
- ▶ **Liegt in der Anwendungsschicht**
  - ▶ 7. Schicht im ISO/OSI Modell
- ▶ **Kommunikation läuft über Text-Nachrichten**
  - ▶ Web-Client sendet HTTP-Requests
  - ▶ Web-Server antwortet mit HTTP-Response
- ▶ **HTTP ist zustandslos**
  - ▶ Server kann sich an vorherige Kommunikation nicht erinnern, kennt den Status des Clients nicht
- ▶ **Nachrichten (Request / Response) bestehen aus:**
  - ▶ Header: Steuerinformationen
  - ▶ Daten: Nutzereingaben oder Dateien

# HTTP – Versionen

---

- ▶ **HTTP 0.9**, 1991:
  - ▶ nur GET implementiert
- ▶ **HTTP 1.0**, [RFC 1945](#):
  - ▶ einfache Authentifikation
  - ▶ neben GET weitere Methoden (POST, HEAD)
  - ▶ Entity-Format: MIME-Typen (MIME - Multipurpose Internet Mail Extensions)
  - ▶ Fehlercodes
- ▶ **HTTP 1.1**, seit 1999, [RFC 2616](#) obsoletes [RFC 2068](#):
  - ▶ unterstützt Virtual Hosting → mehrere virtuelle Domains unter gleicher IP-Adresse,
  - ▶ Verbindungsmanagement → Bezug mehrerer Ressourcen über eine Verbindung
  - ▶ Weitere Methoden (PUT, DELETE, ...) → einfacher Dateitransfer möglich
- ▶ **HTTP 2**, seit 2015
  - ▶ Übertragung wird beschleunigt
  - ▶ Abwärtskompatibel zu HTTP 1.1

# HTTP – Adressierung

---

- ▶ HTTP-Client adressiert eine Datei
- ▶ Die Datei muss auf dem HTTP-Server liegen

- ▶ Adressierung erfolgt per URL (Uniform Resource Locator):

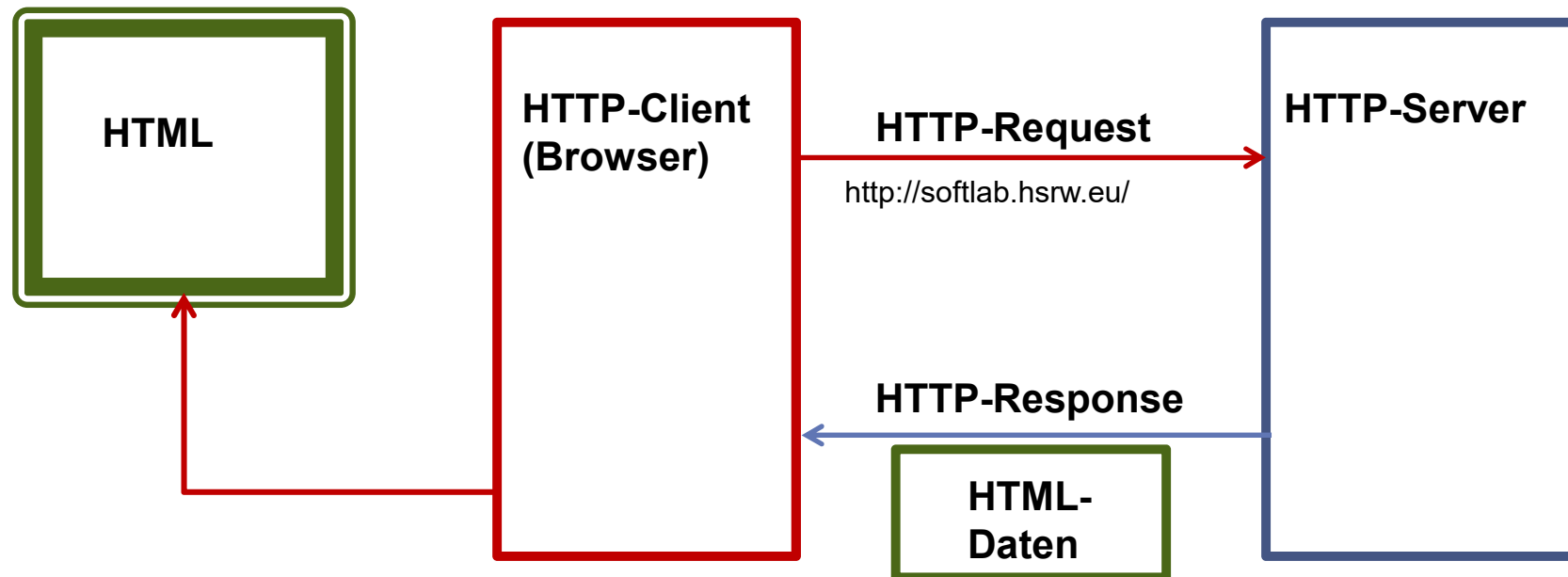
- ▶ [Protokoll://Subdomain.Domainname.Top-Level-Domain:TCP-Port/Pfad/Datei](http://softlab.hochschule-rhein-waal.de:80/trac/lvprogrammierung)



- ▶ Browser erkennt das Protokoll und sendet einen Request an den Domain Name Server (DNS)
- ▶ DNS übersetzt Domainnamen in IP-Adresse
  - ▶ z.B.: <http://193.175.200.208/pfad/datei.html>
- ▶ TCP/IP baut Verbindung zum Server über Standard-Port 80 auf

# HTTP – Ablauf einer Anfrage

---



# HTTP – Ablauf einer Anfrage

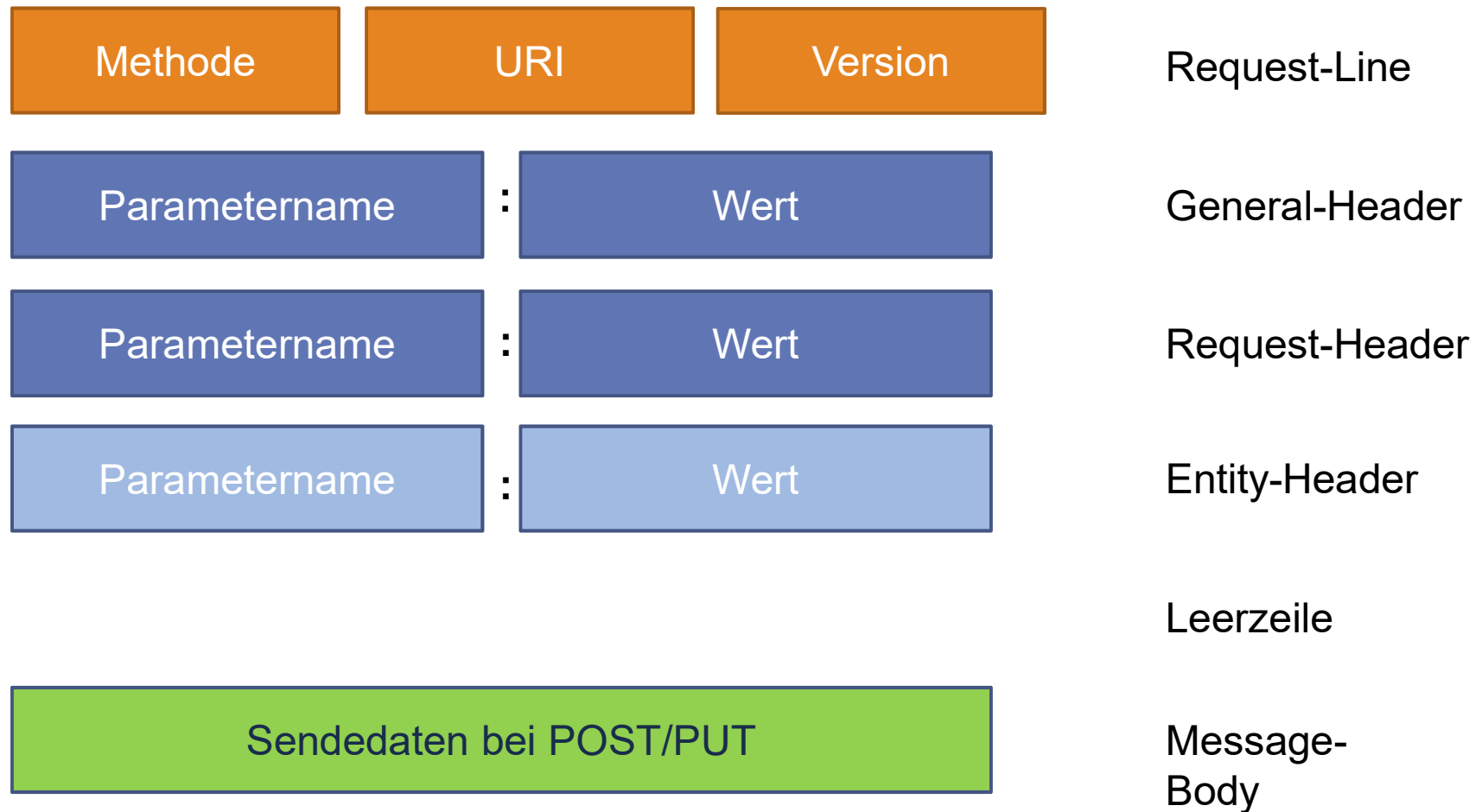
---

- ▶ Ein HTTP Request (Anfrage) wird gesendet
  - ▶ „Schick mir dieses Dokument“ trac/lvprogrammierung
- ▶ Der Server überprüft die Anfrage
  - ▶ Existiert das gewünschte Dokument? Sind Zugriffsberechtigungen gegeben? Im Falle einer dynamisch generierten Datei, leite Anfrage weiter
- ▶ Der Server sendet ein Response (Antwort), z.B.
  - ▶ "OK + Dokument", "Zugriff verweigert", "Seite nicht gefunden", "Serverfehler"
- ▶ Der Browser präsentiert das erhaltene Dokument
  - ▶ HTML (Hypertext Markup Language) zur Strukturierung der Seite
  - ▶ Browser sendet (wenn erforderlich) zusätzliche Anfragen für Bilder etc.
  - ▶ Browser führt clientseitigen Code (z.B. JavaScript) aus

# HTTP – Request

## Aufbau

---



# HTTP – Request

## Beispiel

---

GET /books/index.htm HTTP/1.1

Request-Line

Date: Tue, 15 Feb 2012 08:12:31 GMT  
Connection: Close

General-Header

Accept: image/png, image/gif, \*.\*  
Accept-Language: de, en  
Accept-Encoding: gzip  
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1)  
Host: www.amazon.com  
Referrer: http://www.amazon.com/index.htm

Request-Header

*(Leer bei GET, da kein Body vorhanden)*

Entity-Header



# HTTP – Request

## Beispiel

---

- ▶ Entity-Header enthalten Informationen über den Inhalt
  - ▶ Content-Encoding
  - ▶ Content-Language
  - ▶ Content-Type
  - ▶ Content-Length u.a.
- ▶ Nur Body enthält die eigentlichen Nutzdaten
  - ▶ Z.B. Formulardaten

# HTTP – Methoden

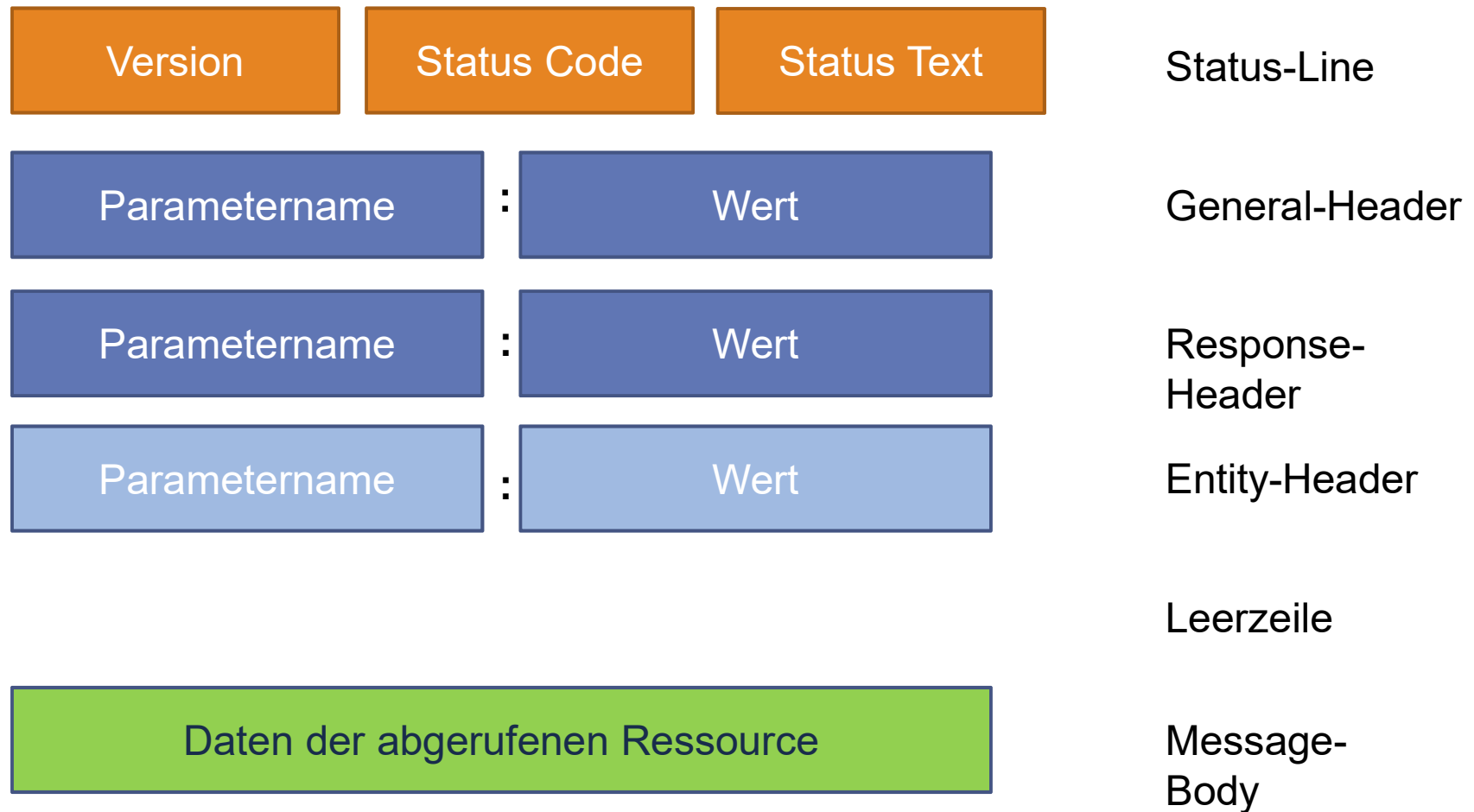
- nicht unterstützt / x erforderlich / o optional

Methode	0.9	1.0	1.1	Beschreibung
GET	x	x	x	Inhalte vom Server anfordern.
HEAD	-	x	x	Weist den Server an, den HTTP-Header, nicht jedoch den Inhalt des Dokuments zu senden. So kann z.B. die Existenz oder Aktualität einer verlinkten Seite geprüft werden.
POST	-	o	o	Inhalte vom Server anfordern und zusätzlich einen Datenblock (z.B. Name-/Wertpaare aus einem Formular) übermitteln. Grundsätzlich können Daten auch mit GET übertragen werden, aber die zulässige Datenmenge bei POST ist größer und nicht in Adresszeile sichtbar.
PUT	-	-	o	Dateien auf den Webserver laden.
DELETE	-	-	o	Löscht die Datei auf dem Server.
OPTIONS	-	-	o	Abruf verfügbarer Kommunikationsoptionen.

# HTTP – Response

## Aufbau

---



# HTTP – Response

## Beispiel

---

HTTP/1.1 200 OK

Status-Line

Date: Tue, 14 Feb 2012 08:13:21 GMT  
Connection: Close

General-Header

Server: Apache/1.3.6 (UNIX)

Response-Header

Last-Modified: Mon, 09 Jan 2012 14:06:11 GMT  
Content-length: 6345  
Content-Type: text/html

Entity-Header

*Leerzeile*

```
<html>
  <head><title>CB CD-Shop</title></head>
  <body><h1>Welcome at my Shop</h1> ....
```

Message-Body

# HTTP – Statuscodegruppen

---

- ▶ **1xx: Informationen**
  - ▶ Die Anfrage wird noch bearbeitet
- ▶ **2xx: Erfolgreiche Operation**
  - ▶ Die Anfrage wurde erfolgreich bearbeitet
- ▶ **3xx: Umleitung**
  - ▶ Weitere Schritte sind erforderlich, um mit der Bearbeitung fortfahren zu können
- ▶ **4xx: Client-Fehler**
  - ▶ Die Anfrage enthält eine fehlerhafte Syntax oder kann nicht ausgeführt werden
- ▶ **5xx: Server-Fehler**
  - ▶ Der Server kann die Anfrage nicht bearbeiten

# HTTP – Statuscodes

---

Code	Bedeutung
200 OK	Anfrage OK, hier ist Response
301 Moved Permanently	Angeforderte Ressource hat eine neue Adresse
304 Not Modified	Dokument wurde seit letztem Zugriff nicht geändert
400 Bad Request	Syntaxfehler in der Anfrage
401 Unauthorized	Client ist nicht berechtigt, auf Ressource zuzugreifen
403 Forbidden	Zugriff verweigert
404 Not Found	Das Dokument existiert nicht
500 Internal Server Error	Ein unerwarteter Server-Fehler ist aufgetreten