VEGARD IT

Mit `curl` auf Fehlersuche

Mehr als HTTP & APIs

Christoph Stoettner

2026-03-28

Christoph Stoettner (stoeps)

stoeps.de

@stoeps@infosec.exchange

m @stoeps:nope.chat

- seit 30 Jahren was mit Computern
  - Amiga, OS/2, Linux
  - Beruflich auch Windows (wenn es sein muss)
- Linux / OSS seit 1994/1995
  - Linux Kernel < 1.0
  - Slackware
- mag `vi`, `vim`, `neovim`
- Was ist `nano`?

Wenn ein Prozess Excel enthält, ist der Prozess kaputt.

Chemnitzer Linuxtage 2026

Danke

- Orga Team und Helfern Chemnitzer Linuxtage
- Familie
- Tobi von https://toheine.net für das `asciinema` Skript

Warum dieser Vortrag?

- Troubleshooting in
  - Bare Metal
  - Virtuellen Maschinen
  - Containern (Podman, Docker, Kubernetes usw.)
- `telnet` seit Jahren nicht mehr Standard (und das ist gut so)
- `netcat` wird von IDS, IPS und SIEM protokoliert oder geblockt

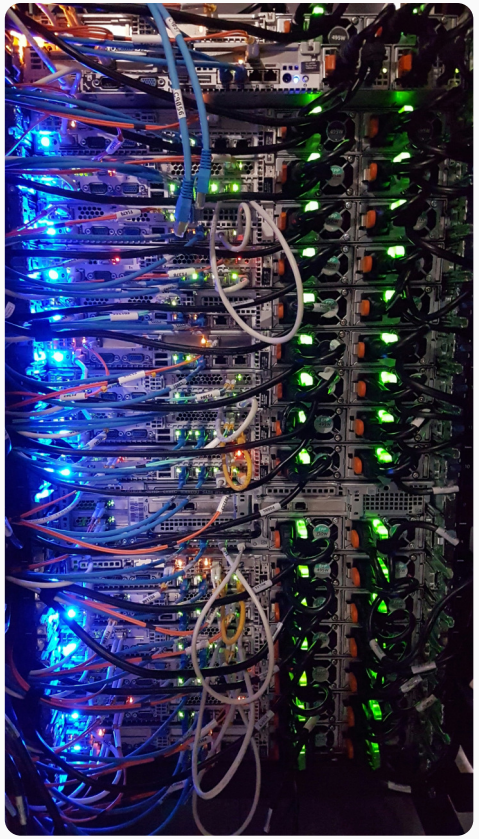


Abbildung 1: Massimo Botturi [1]

Lösung

- Nachinstallieren?
  - Container?
  - Systeme ohne Internet
- Ist der Port offen
  - SMTP
  - IMAP
  - POP3

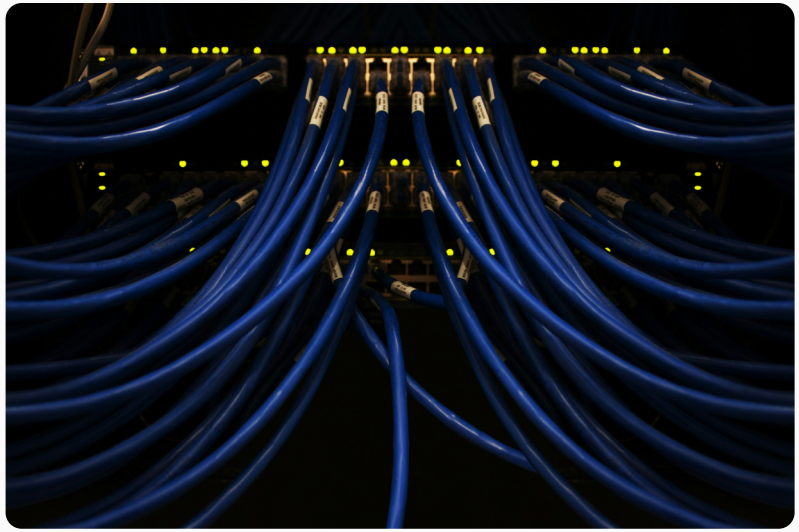


Abbildung 2: Scott Rodgerson [2]

- Leitet der Reverse Proxy zum richtigen Server

Verbindungen oft nur noch verschlüsselt

Historisch war `telnet` die Lösung

- Port offen

```bash
telnet example.com 80
Trying 104.18.26.120...
Connected to example.com.
Escape character is '^]'.
```

- Port geschlossen

```bash
telnet example.com 81
Trying 104.18.27.120...
 telnet: connect to address
104.18.27.120: Connection timed out
```

- mit ausreichend Wissen über ein Protokoll konnte man den Server testen
- RFC lesen hilft auch

SMTP / HTTP

```bash
telnet 127.0.0.1 25
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
220 8350e8d948d9 ESMTP Postfix (Debian)
ehlo example.org
mail from: stoeps@example.org
250 2.1.0 Ok
rcpt to: stoeps@stoeps.de
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
subject: Test
Some text.
.
250 2.0.0 Ok: queued as E9C1D4575AE2
```

```bash
telnet google.com 80
Trying 172.217.168.78...
Connected to google.com.
Escape character is '^]'.
GET /index.html
HTTP/1.0 200 OK
Date: Sat, 28 Feb 2026 15:06:22 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html;
charset=ISO-8859-1
```

`curl` kann auch `telnet`

- `curl telnet://192.168.11.1`
- Zum Prüfen ob Port offen → OK
- Ich kenne einen Podcast, bei dem kann man Feedback über `telnet` geben
  - Getestet und funktioniert

Photo: [3]

Aber `curl` kann viel mehr

Protokolle

`DICT` `FILE` `FTP` `FTPS` `GOPHER` `GOPHERS` `HTTP` `HTTPS` `IMAP`
`IMAPS` `LDAP` `LDAPS` `MQTT` `MQTTS` `POP3` `POP3S` `RTMP` `RTMPS`
`RTSP` `SCP` `SFTP` `SMB` `SMBS` `SMTP` `SMTPS` `TELNET` `TFTP` `WS` `WSS`

Aber `curl` kann viel mehr

Protokolle

`DICT` `FILE` `FTP` `FTPS` `GOPHER` `GOPHERS` `HTTP` `HTTPS` `IMAP`
`IMAPS` `LDAP` `LDAPS` `MQTT` `MQTTS` `POP3` `POP3S` `RTMP` `RTMPS`
`RTSP` `SCP` `SFTP` `SMB` `SMBS` `SMTP` `SMTPS` `TELNET` `TFTP` `WS` `WSS`

Proxies

`SOCKS4` `SOCKS5` `HTTP` `HTTPS` `HTTP/1` `HTTP/2` `tunneling`
`unix sockets` `haproxy` `SOCKS+HTTP chain`

Chemnitzer Linuxtage 2026

Aber `curl` kann viel mehr

Protokolle

`DICT` `FILE` `FTP` `FTPS` `GOPHER` `GOPHERS` `HTTP` `HTTPS` `IMAP`
`IMAPS` `LDAP` `LDAPS` `MQTT` `MQTTS` `POP3` `POP3S` `RTMP` `RTMPS`
`RTSP` `SCP` `SFTP` `SMB` `SMBS` `SMTP` `SMTPS` `TELNET` `TFTP` `WS` `WSS`

Proxies

`SOCKS4` `SOCKS5` `HTTP` `HTTPS` `HTTP/1` `HTTP/2` `tunneling`
`unix sockets` `haproxy` `SOCKS+HTTP chain`

Auth

`Basic` `Plain` `Digest` `CRAM-MD5` `SCRAM-SHA` `NTLM` `Negotiate`
`Kerberos` `Bearer` `AWS Sigv4` `SASL` `.netrc`

Chemnitzer Linuxtage 2026

Ein paar Non-HTTP Beispiele

SMTP mit `curl`

```bash
curl --url "smtp://smtp.example.com:587" \
  --ssl-reqd \
  --mail-from "john.doe@example.com" \
  --mail-rcpt "musterfrau@example.com" \
  --upload-file "email" \
  --insecure -n
```

```email
From: John Doe <John.Doe@example.com
To: musterfrau@example.com
Subject: Testing SMTP Mails with curl

Hi,
just wanted to test.

Gallia est omnis divisa i n partes tres, quarum.
```

POP3 / IMAP mit `curl`

```bash
curl pop3(s)://example.com
curl imap(s)://example.com
```

- Check Inbox

```bash
curl imaps://imap.example.com/  -X 'STATUS INBOX (MESSAGES)' -n
* STATUS "INBOX" (MESSAGES 1)
```

- Lesen der 1. Mail

```bash
curl 'imaps://imap.example.com/INBOX;MAILINDEX=1' -n
```

IMAP mehr Details

- Ordner-Liste

```url
imap://user:password@mail.example.com
```

- Ordner-Liste in der INBOX

```url
imap://user:password@mail.example.com/INBOX
```

- INBOX auswählen und Nachricht mit UID=1 anzeigen

```url
imap://user:password@mail.example.com/INBOX/;UID=1
```

- INBOX auswählen und die erste Nachricht anzeigen

IMAP mehr Details

```url
imap://user:password@mail.example.com/INBOX/;MAILINDEX=1
```

- Zeige den Textteil der 3. Nachricht in der INBOX

```url
imap://user:password@mail.example.com/INBOX/;UID=3/;SECTION=TEXT
```

- Zeige die ersten 1024 Octets der 4. Nachricht in der INBOX

```url
imap://user:password@mail.example.com/INBOX/;UID=4/;PARTIAL=0.1024
```

- Prüfe auf neue Nachrichten in der Inbox

```url
imap://user:password@mail.example.com/INBOX?NEW
```

IMAP mehr Details

- Suche in der INBOX nach Nachrichten die `shadows` im Subject enthalten

```url
imap://user:password@mail.example.com/INBOX?SUBJECT%20shadows
```

LDAP Query mit `curl`

```bash
curl -n 'ldap://cnx-ds.stoeps.home/dc=stoeps,dc=home?givenname,sn,mail?sub?
(givenname=Christoph)'
DN: uid=stoeps,dc=stoeps,dc=home
        givenName: Christoph
        sn: Stoettner
        mail: stoeps@stoeps.home
```

$$\underbrace{\text{ldap(s)://cnx-ds.stoeps.home/}}_{\text{protocol://host}}\underbrace{\text{dc=stoeps,dc=home}}_{\text{base DN}}?\underbrace{\text{cn,mail}}_{\substack{\text{kommagetrennt}\\\text{Attribute}}}?\underbrace{\text{base}}_{\substack{\text{scope}\\\text{base|one|sub}}}?\underbrace{\text{(objectclass=*)}}_{\text{filter}}$$

Figure 1: Struktur der LDAP URL, Rückgabe CN

$$\underbrace{\text{ldap(s)://cnx-ds.stoeps.home/}}_{\text{protocol://host}}\underbrace{\text{dc=stoeps,dc=home}}_{\text{base DN}}?\underbrace{}_{\substack{\text{all attributes}}}?\underbrace{\text{sub}}_{\text{base|one|sub}}?\underbrace{\text{(\&(objectclass=person)(givenname=Christoph))}}_{\text{filter}}$$

Figure 2: LDAP URL, Rückgabe alle Attribute

SFTP / SCP

```bash
curl sftp://example.com/somefile.zip -u user
curl scp://example.com/somefile.zip -u user
curl sftp://example.com/ -u user
curl sftp://example.com/~/somefile.zip -u user
```

- `-T`, `--upload-file <file>` Upload einer lokalen Datei
- Download mit Option `-O` oder `-o` Dateiname
- `--key <key>` SSH key zur Authentifizierung

`--key` auth ist komisch implementiert. Bei Passwort geschützten Keys, hat die Verbindung bei mir nur mit `ssh-agent` geklappt, Passwort-Prompt mit Enter schliessen.

MQTT

- Subscription (Terminal 1)
  - `-o` schreibt die Antworten in eine Datei

```bash
curl -N mqtt://broker.emqx.io/curl/test
```

- Publish (Start in Terminal 2 - Ausgabe sichtbar in 1)

```bash
curl -v -d "Hello from curl" mqtt://broker.emqx.io/curl/test
```

- Verbindungstest

```bash
curl -v mqtt://broker.emqx.io/test
```

MQTT - Demo

```bash
$ curl -N mqtt://broker.emqx.io/curl/test -o output.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    30  100    30    0     0      3      0  0:00:10  0:00:08  0:00:02     5

$ cat output.txt
       curl/testHello from CLT 2026%
```

```bash
$ curl -v -d "Hello from CLT 2026" mqtt://broker.emqx.io/curl/test
* Host broker.emqx.io:1883 was resolved.
* Connected to broker.emqx.io (44.232.241.40) port 1883
* Using client id 'curlxcPQKT2M'
> MQTT<
      curlxcPQKT2Mmqtt_doing: state [0]
* mqtt_doing: state [0]
<  mqtt_doing: state [2]
< 0     curl/testHello from CLT 2026shutting down connection #0``
```

DICT

- DICT ist ein Protokoll für Wörterbuchabfragen
- Aliase für `m` sind `match` und `find`
- Aliase für `d` sind `define` und `lookup`

```bash
curl dict://dict.org/d:Chemnitz
220 dict.dict.org dictd 1.12.1/rf on Linux 4.19.0-10-amd64 <auth.mime>
<671773095.13632.1774187333@dict.dict.org>
250 ok
150 1 definitions retrieved
151 "chemnitz" wn "WordNet (r) 3.0 (2006)"
Chemnitz
    n 1: a city in east central Germany; formerly called Karl-Marx-
         Stadt until 1990; noted for textile manufacturing [syn:
         {Chemnitz}, {Karl-Marx-Stadt}]
.
...
```

Trivial File Transfer Protocol (TFTP)

- Download

```bash
curl -O tftp://localhost/file.boot
```

- Upload Beispiel

```bash
curl -k -u admin:secret -H "Accept:pplication/json" \
     --cacert /etc/puppetlabs/puppet/ssl/certs/ca.pem \
     --cert /etc/puppetlabs/puppet/ssl/certs/foreman.localdomain.pem \
     --key /etc/puppetlabs/puppet/ssl/private_keys/foreman.localdomain.pem
     -X POST https://foreman:8443/tftp/00:11:22:33:44:55 \
     --data "syslinux_config=cat data.json"
```

SMB

- Upload und Download
- Kann keine Verzeichnislisten anzeigen
- Unterstützt nur SMB v1
- Authentifizierung benutzt NTLMv1 und nicht v

Abbildung 4: Tadas Sar [4]

„NTLMv1 und SMBv1 sind unsicher und sollten nicht mehr verwendet werden!“

Gopher

Gopher wurde 1991 unter der Leitung von Mark P. McCahill an der Universität von Minnesota entwickelt und ähnelt dem World Wide Web (WWW) in einem frühen Zustand.

```bash
curl 'gopher://quux.org/'
iWelcome to gopher at quux.org! fake    (NULL)  0
i       fake    (NULL)  0
iThis server has a lot of information of historic interest,     fake    (NULL)  0
ifunny, or just plain entertaining -- all presented in Gopher.  fake    (NULL)  0
...
0About This Server      /About This Server.txt  gopher.quux.org 70      +
1Archives       /Archives       gopher.quux.org 70      +
1Books  /Books  gopher.quux.org 70      +
...
0What's New     /whatsnew.txt   gopher.quux.org 70      +
```

Gopher Verzeichnisse `/1/`

```bash
curl 'gopher://quux.org/1/Archives'
1Mailing Lists  /Archives/Mailing Lists gopher.quux.org 70      +
0Project Gutenberg Online Books /Archives/gutenberg     gopher.quux.org 70
1Usenet Archives from 1981      /Archives/usenet-a-news gopher.quux.org 70      +
1Wiretap Electronic Documents   /Archives/mirrors/wiretap.area.com
gopher.quux.org       70
1mirrors        /Archives/mirrors       gopher.quux.org 70      +
```

Gopher Dokumente `/0/`

```bash
curl gopher://quux.org:70/0/About%20This%20Server.txt
```

```
Welcome to the gopher server at quux.org!

This is one of the world's few maintained, modern gopher servers.  On it,
you will find a huge collection of information, files, software, archives,
...
```

```bash
curl 'gopher://quux.org/0/whatsnew.txt'
```

```
NEW THINGS ON QUUX.ORG
----------------------
April 2, 2002
 * Major expansion.  Added EFF mirror, the US Congress Bills,
  expanded many other mirrors, updated textfiles mirror.

 * Placed much more material in proper files.

 * This site now contains over 200,000 files.
```

HTTP

Downloads / Output

- Jede heruntergeladene Datei braucht ein Ziel!
  - `stdout` oder eine Datei
- `-o`, `--output <Dateiname>`
- `-O`, `--remote-name`, speichere mit dem Remote-Dateinamen
- `--remote-name-all` setzt `-O` für alle URLs

Progress

- `-s`, `--silent`, `--no-progress-meter`
- `-#`, `--progress-bar`

```
presentations/demo/clt2026-casts on  main [!?]
⬢ [fedora-stoeps] ❯ curl --progress-bar -l -O https://mirror.netzwerge.de/almalinux/10.1/isos/x86_64/AlmaLinux-10.1-x86_64-boot.iso
###################                                                                                                15.6%
```

Abbildung 5: Progress Bar beim Download

```
presentations/demo/clt2026-casts on  main [!?] took 52s
⬢ [fedora-stoeps] ❯ curl -l -O https://mirror.netzwerge.de/almalinux/10.1/isos/x86_64/AlmaLinux-10.1-x86_64-boot.iso
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  8  926M    8 75.4M    0     0  13.4M      0  0:01:08  0:00:05  0:01:03 14.8M
```

Abbildung 6: Normaler Download

URLs and Optionen

- `curl` akzeptiert mehrere URLs pro Kommando
- erstes `-o` enthält den Dateinamen der ersten URL!

```
curl -o file1 -o file2 https://example.com/file1 https://example.com/file2
```

```
curl -o file1 https://example.com/file1 https://example.com/file2 -o file2
```

```
curl -o file1 \
     https://example.com/file1 \
     -o file2 \
     https://example.com/file2
```

Mehrere Dateien herunterladen

```
presentations/demo/clt2026-casts on  main [!?]
⬢ [fedora-stoeps] ❯ curl -O https://mirror.netzwerge.de/almalinux/10.1/isos/x86_64/AlmaLinux-10.1-x86_64-boot.iso --next -O https://
mirror.netzwerge.de/almalinux/10.1/isos/x86_64/AlmaLinux-10.1-x86_64-minimal.iso
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  926M  100  926M    0     0  15.6M      0  0:00:59  0:00:59 --:--:-- 16.0M
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
 32 1464M   32  471M    0     0  14.6M      0  0:01:40  0:00:32  0:01:08 14.7M
```

Abbildung 7: Download von 2 ISO Dateien

```
presentations/demo/clt2026-casts on  main [!?] took 9s
⬢ [fedora-stoeps] ❯ curl -O https://mirror.netzwerge.de/almalinux/10.1/isos/x86_64/AlmaLinux-10.1-x86_64-boot.iso --next -O https://
mirror.netzwerge.de/almalinux/10.1/isos/x86_64/AlmaLinux-10.1-x86_64-minimal.iso --parallel
DL% UL%  Dled  Uled  Xfers  Live Total     Current  Left    Speed
 57 --  1385M     0     2     1   0:03:13  0:01:37  0:01:21 12.3M
```

Abbildung 8: Paralleles Herunterladen

URL encode

- `trurl` zur Konvertierung
- oder `--url-query` Kommandozeilen Option

```bash
curl 'http://example.com/index.php?name=Christoph Stöttner' # malformed
curl http://example.com/index.php\?name=Christoph+St%c3%b6ttner # urlencoded
curl http://example.com/index.php --url-query 'name=Christoph Stöttner'
```

```
GET /index.php?name=Christoph+St%c3%b6ttner HTTP/1.1
Host: example.com
User-Agent: curl/8.15.0
Accept: */*
```

Abbildung 9: `--url-query` automatisches urlencode

HTTP Anfragen

- Standard HTTP Request (`-X GET` ist Default)

```bash
curl -X GET https://stoeps.de
```

```
presentations/demo/clt2026-casts on  main [!?] took 2m39s
 [fedora-stoeps] ❯ curl -X GET https://stoeps.de -L

<!doctype html><html lang=en><head><meta charset=utf-8><meta http-equiv=x-ua-compatible content="ie=edge; chrome=1"><meta name=viewp
ort content="width=device-width,initial-scale=1,maximum-scale=5"><title>stoeps.de - Christoph Stoettner's Blog &#183; stoeps' blog</
title><meta name=geo.position content="49.642538;8.638950"><meta name=geo.region content="DE"><meta name=geo.placename content="Hepp
enheim, Germany"><meta name=ICBM content="49.642538, 8.638950"><meta name=DC.title content="stoeps' blog"><meta name=author content=
```

Abbildung 10: Webseite abfragen

SSL Zertifikate

- Selfsigned Zertifikate, Hostname mismatch, abgelaufene Zertifikate erzeugen Fehler
- `-k` alle Zertifikatsfehler ignorieren
- `--ca-native` System Key

```bash
$ curl https://expired.badssl.com/
curl: (60) SSL certificate problem: certificate has expired
More details here: https://curl.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the webpage mentioned above.
```

Abgelaufenes SSL Zertifikat ignorieren

```bash
curl https://expired.badssl.com/ -k
```

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>expired.badssl.com</title>
  <link rel="stylesheet" href="/style.css">
  <style>body { background: red; }</style>
</head>
<body>
<div id="content">
  <h1 style="font-size: 12vw;">
    expired.<br>badssl.com
  </h1>
</div>
</body>
</html>
```

SSL Protokolle testen

- nur TLS 1.1
  - `curl https://stoeps.de -v --tlsv1.1 --tls-max 1.1`
- nur TLS 1.2
  - `curl https://stoeps.de -v --tlsv1.2 --tls-max 1.2`
- nur TLS 1.3
  - `curl https://stoeps.de -v --tlsv1.3 --tls-max 1.3`
- minimal TLS 1.1, maximal TLS 1.2
  - `curl https://stoeps.de -v --tlsv1.1 --tls-max 1.2`
- mindestens TLS 1.0
  - `curl https://stoeps.de -v --tlsv1.0`

HTTP Status 3xx - Umleitung

- HTTP Calls mit Response Code 3xx (Umleitung)

```bash
curl -XGET  https://chemnitzer.linux-tage.de
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/
strict.dtd">
<html><head>
<title>302 Found</title>
</head><body>
<h1>Found</h1>
<p>The document has moved <a href="https://chemnitzer.linux-tage.de/2026">here</
a>.</p>
</body></html>
```

3xx Umleitung zu neuer Location folgen

- `-L`, `--location`

```bash
curl -XGET  https://chemnitzer.linux-tage.de --location
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="de" xml:lang="de"
      dir="ltr">
<head>
    <meta content="width=device-width, initial-scale=1.0, maximum-scale=3.0"
name="viewport"/>
<meta name="description" content="Die Chemnitzer Linux-Tage sind eine Veranstaltung
rund um das Thema Linux und Open Source für jedermann, die Linux-Nutzer, Insider und
Unternehmen zusammenbringt."/>
<meta name="keywords"
      content="Linux, Linux-Tag, Linux-Tage, Chemnitzer Linux-Tag, Chemnitzer Linux-
Tage 2026, Chemnitz, Open Source, 2026, Insider, Firmen"/>
<meta name="robots" content="index, follow"/>
```

Kubernetes Ingress Konfiguration überprüfen

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
...
spec:
  ingressClassName: cnx-ingress-traefik
  rules:
  - host: cnx8-db2.stoeps.home
    http:
      paths:
      - backend:
          service:
            name: orient-web-client
            port:
              number: 8000
        path: /social
        pathType: Prefix
        ...
```

- `cnx8-db2.stoeps.home` vorgelagerter LB
- Kubernetes Nodes haben anderen Namen

`--connect-to`

- Request für HOST1:PORT an HOST2:PORT2 senden
- Syntax `curl --connect-to HOST1:PORT1:HOST2:PORT2 ...`

Hostname	Funktion
cnx8-db2-cp	Kubernetes Node
cnx8-db2	Load Balancer

- Original Call: `https://cnx8-db2.stoeps.home/cnxadmin`
- Wir wollen ohne LB testen

```bash
curl -kL https://cnx8-db2-cp.stoeps.home:32443/cnxadmin
# 404
```

`--connect-to`

```bash
curl -kL \
     --connect-to cnx8-db2.stoeps.home:443:cnx8-db2-cp.stoeps.home:32443 \
     https://cnx8-db2.stoeps.home/cnxadmin
# Page delivered
```

- mit entsprechenden Headern auch ohne `--connect-to` möglich
- einfachste Lösung um Verbindungen ohne LB zu testen

Namensauflösung

```conf
NameVirtualHost *:80
<VirtualHost *:80>
    ServerName www.example.com
    DocumentRoot "/www/domain"
</VirtualHost>

<VirtualHost *:80>
    ServerName other.example.com
    DocumentRoot "/www/otherdomain"
</VirtualHost>
```

- Ohne `/etc/hosts` Änderung testen?

--resolve

- Normale Seite von www.example.com

```bash
curl http://www.example.com
<!doctype html><html lang="en"><head><title>Example Domain</title><
```

- Lokalen Webserver mit www.example.com aufrufen

```bash
curl --resolve www.example.com:80:10.0.22.90 http://www.example.com
<!DOCTYPE html>
...
    <title>domain</title>

curl --resolve other.example.com:80:10.0.22.90 http://other.example.com
<!DOCTYPE html>
...
    <title>otherdomain</title>
```

DNS over HTTPS

- `--doh-url`
- Angabe eines Hosts für DNS over HTTPS

```bash
curl --doh-url https://cloudflare-dns.com/dns-query https://chemnitzer.linux-tage.de
```

- Auflösung des DoH Servers mit `--resolve` ändern

```bash
curl --doh-url https://doh.example --resolve doh.example:443:192.0.2.1 https://example.com
```

Websockets – alter Weg

```bash
curl -i -N \
     -H "Connection: Upgrade" \
     -H "Upgrade: websocket" \
     -H "Host: echo.websocket.org" \
     -H "Origin: https://www.websocket.org" \
     https://echo.websocket.org

...
--------------------------------------------------------------------
         __         __   _                    _          _
         \ \    / /__| |__  ___ ___  __| |_____| |_
          \ \/\/ / -_) '_ \(_-</ _ \/ _| / / -_)  _|
           \_/\_/\___|_.__//__/\___/\__|_\_\___|\__|

  WebSocket UI: https://echo.websocket.org/.ws
  SSE: https://echo.websocket.org/.sse
  Learn more: https://websocket.org/tools/websocket-echo-server
--------------------------------------------------------------------
```

Websockets mittels `wss://`

- `Connection: upgrade` automatisch

```bash
curl --no-progress-meter --no-buffer wss://echo.websocket.org -v
```

```
< upgrade: websocket
< connection: Upgrade
< sec-websocket-accept: kYxJcvZB0Ry3mYSjifDWmKqN22o=
< date: Thu, 26 Mar 2026 18:05:06 GMT
< server: Fly/58128dbb4 (2026-03-25)
< via: 1.1 fly.io, 1.1 fly.io
< fly-request-id: 01KMNN4JPHH9XNCNHT1F3DN95E-fra
<
* [WS] Received 101, switch to WebSocket; mask 111e9f0d
Request served by 4d896d95b55478
```

Kopieren aus den Browser Web Developer Tools

- Web Developertools `F12`
- Filtern nach Request

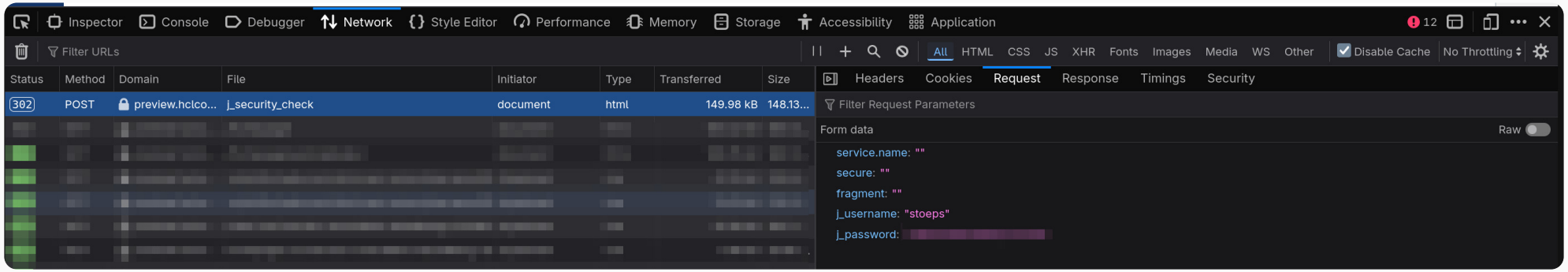


Abbildung 11: Firefox Screenshot POST

Kopieren aus den Browser Web Developer Tools

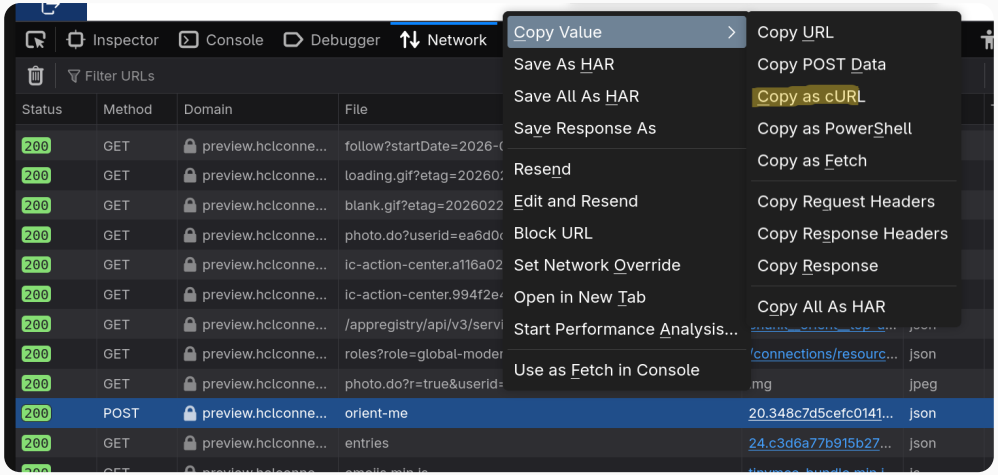


Abbildung 12: Copy as `curl`

- Alle Header und Cookies werden kopiert
  - nicht sharen (solange Cookies noch nicht abgelaufen sind)

Kopieren aus den Browser Web Developer Tools

```bash
curl 'https://other.example.com/social/auth/token' \
  --compressed \
  -H 'User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:148.0) Gecko/20100101
Firefox/148.0' \
  -H 'Accept: */*' \
  -H 'Accept-Language: en-US,en;q=0.9,de-DE;q=0.8' \
  -H 'Accept-Encoding: gzip, deflate, br, zstd' \
  -H 'Referer: https://other.example.com/homepage/web/updates/' \
  -H 'Connection: keep-alive' \
  -H 'Cookie: lcLang=en_us; JSESSIONID=...' \
  -H 'Sec-Fetch-Dest: empty' \
  -H 'Sec-Fetch-Mode: cors' \
  -H 'Sec-Fetch-Site: same-origin' \
  -H 'If-Modified-Since: Thu, 26 Mar 2026 06:04:12 GMT' \
  -H 'If-None-Match: W/"2-nOO9QiTIwXgNtWtBJezz8kv3SLc"' \
  -H 'Priority: u=4'
```

Authentifizierung und Konfiguration

Config-Datei

- Default: `~/.curlrc`
- `-K [file]` or `--config [file]`

```
~/.curlrc
# Enable verbose output for debugging
verbose

# Suppress progress meter but still show error messages
silent

# Set default user agent
user-agent = "Mozilla/5.0 (compatible; ExampleBot/1.0)"

# Follow redirects
location
```

Authentifizierung

- Basic Auth
  - `--user, -u username:passwort`
- Digest

```bash
curl --digest --user stoeps:pass http://example.com/
```

- Negotiate (Spnego/Kerberos)

```bash
curl --negotiate --user stoeps:pass http://example.com/
```

- NTLM

```bash
curl --ntlm --user stoeps:pass http://example.com/
```

Abbildung 13: Markus Spiske [5]

Authentifizierung `.netrc`

- Pfad
  - Global in `~/.netrc`
  - Oder mit Umgebungsvariable `NETRC`
  - `curl --netrc-file <filename>`
- Aktivieren mit `-n` bzw. `--netrc`
- Format:

```
default login anonymous password user@site
machine-name login anonymous password user@site
```

```
machine-name
login anonymous
password password
```

In den Beispielen v.a. um keine Passwörter anzuzeigen.

Form based Authentication

- `-X POST`

```bash
curl 'https://other.example.com/homepage/j_security_check' \
  -X POST \
  -H 'User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:148.0) Gecko/20100101
Firefox/148.0' \
  -H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8' \
  -H 'Content-Type: application/x-www-form-urlencoded' \
  --data-raw
'service.name=&secure=&fragment=&j_username=my_username&j_password=my_password'
```

Cookies

- Anmeldung und Cookies in Datei schreiben `-c, --cookie-jar`

```bash
curl -v 'https://preview.hclconnections.net/homepage/j_security_check' \
  -H @header.txt \
  --data @data.txt \
  --cookie-jar cookiestxt
```

- Request mit vorher erstellter Cookie Datei `-b, --cookie`

```bash
curl https://preview.hclconnections.net/profiles/photo.do\?
userid\=2a498bd0-3d7f-103c-9a59-c5858c984eb9 \
     -I \
     --cookie cookiestxt
```

Cookies aktualisieren

- Oder kombiniert, `curl` aktualisiert dann Cookies im jar

```bash
curl https://preview.hclconnections.net/profiles/photo.do\?
userid\=2a498bd0-3d7f-103c-9a59-c5858c984eb9 \
     -I \
     --cookie-jar cookiestxt \
     --cookie cookiestxt
```

```cookiestxt
# Netscape HTTP Cookie File
# https://curl.se/docs/http-cookies.html
# This file was generated by libcurl! Edit at your own risk.

#HttpOnly_preview.hclconnections.net  FALSE / TRUE  0 JSESSIONID
0000FMswii6vxnVO7Bmg_O8Y10c:1ij6o8eh3
preview.hclconnections.net  FALSE / TRUE  0 lcLang  en_us
#HttpOnly_.hclconnections.net TRUE  / TRUE  0 LtpaToken2  8qQux2lqN8e7...
```

`curl` in Shellskripten

- Troubleshooting eines Shellscripts

```bash
curl -o testfile https://www.example.com/downloadtest.php
```

- In Ansible (stark verkürzt)

```yaml
- name: Download file
  shell: /tmp/download.sh
  register: command_result
  failed_when: command_result.stderr != ''
```

StdErr

- `curl` sendet nur Daten (data) an `stdout`

```bash
curl -o testfile https://www.example.com/downloadtest.php 1>/dev/null
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 19840  100 19840    0     0  61043      0 --:--:-- --:--:-- --:--:-- 61234
```

- besser

```yaml
failed_when: command_result.rc != 0
```

- https://github.com/curl/curl/issues/12416
  - This is a fundamental key part of and design decision in curl since day one. Only data is sent to stdout.

Links:

- curl.se
- Daniel Stenberg
- GitHub curl/curl
- Everything curl
- Mastering curl

Fragen

@stoeps@infosec.exchange
stoeps.de
@stoeps:nope.chat

Chemnitzer Linuxtage 2026

Quellen und Links

[1] Massimo Botturi (https://unsplash.com/@wildmax), 2020. [Online]. Verfügbar unter: https://unsplash.com/photos/electronic-wire-lot-zFYUsLk_50Y

[2] Scott Rodgerson (https://unsplash.com/@scottrodgerson), 2023. [Online]. Verfügbar unter: https://unsplash.com/photos/a-bunch-of-blue-wires-connected-to-each-other-PSpf_XgOM5w

[3] Patrick (https://unsplash.com/@pf91_photography), 2021. [Online]. Verfügbar unter: https://unsplash.com/photos/red-and-silver-multi-tool-2NuEzrmD2xQ

[4] Tadas Sar (https://unsplash.com/@stadsa), 2018. [Online]. Verfügbar unter: https://unsplash.com/photos/black-laptop-computer-keyboard-in-closeup-photo-T01GZhBSyMQ

[5] Markus Spiske (https://unsplash.com/@markusspiske), 2020. [Online]. Verfügbar unter: https://unsplash.com/photos/text-nBwhHm69x4I

Das `curl` Projekt

Daten und Fakten

- Ursprünglich `HttpGet`, `urlget`
- Rafael Sagula
- Daniel Stenberg (mehr als 20.000 Commits)
- Mehr als 1100 Contributor
- 40.800 Stars bei GitHub
- 7.100 Forks

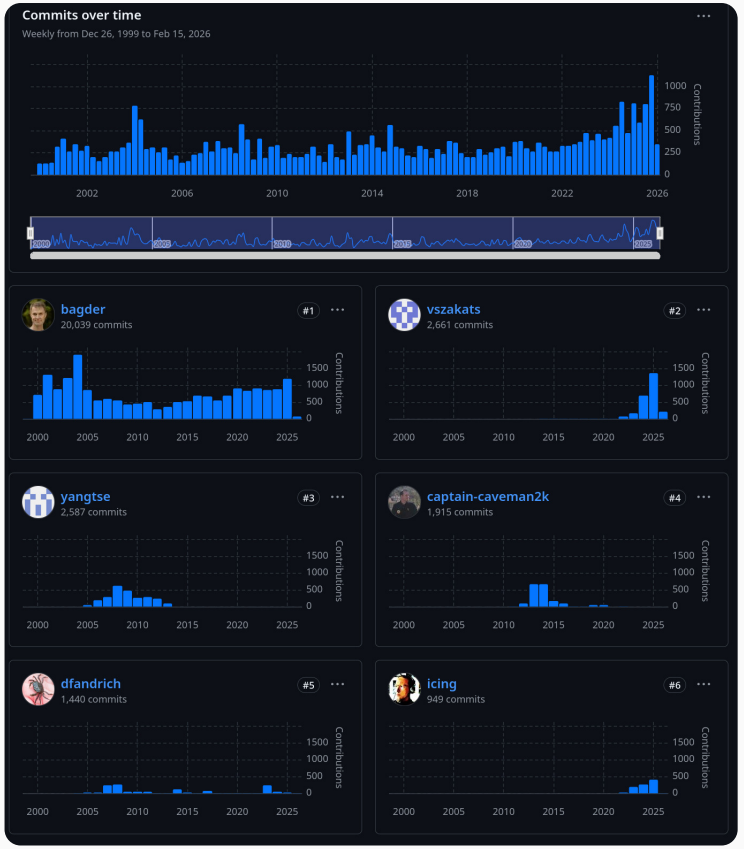
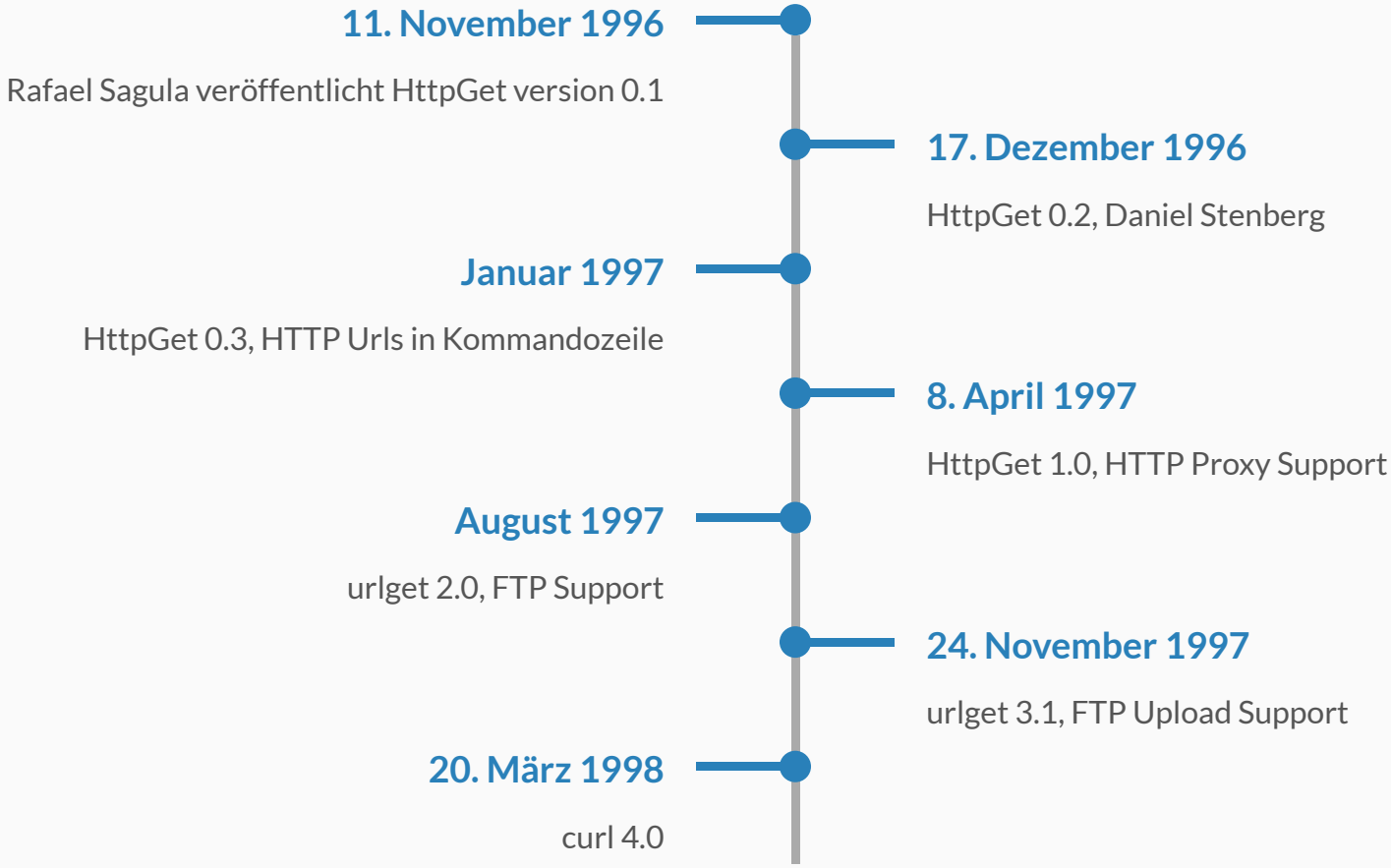
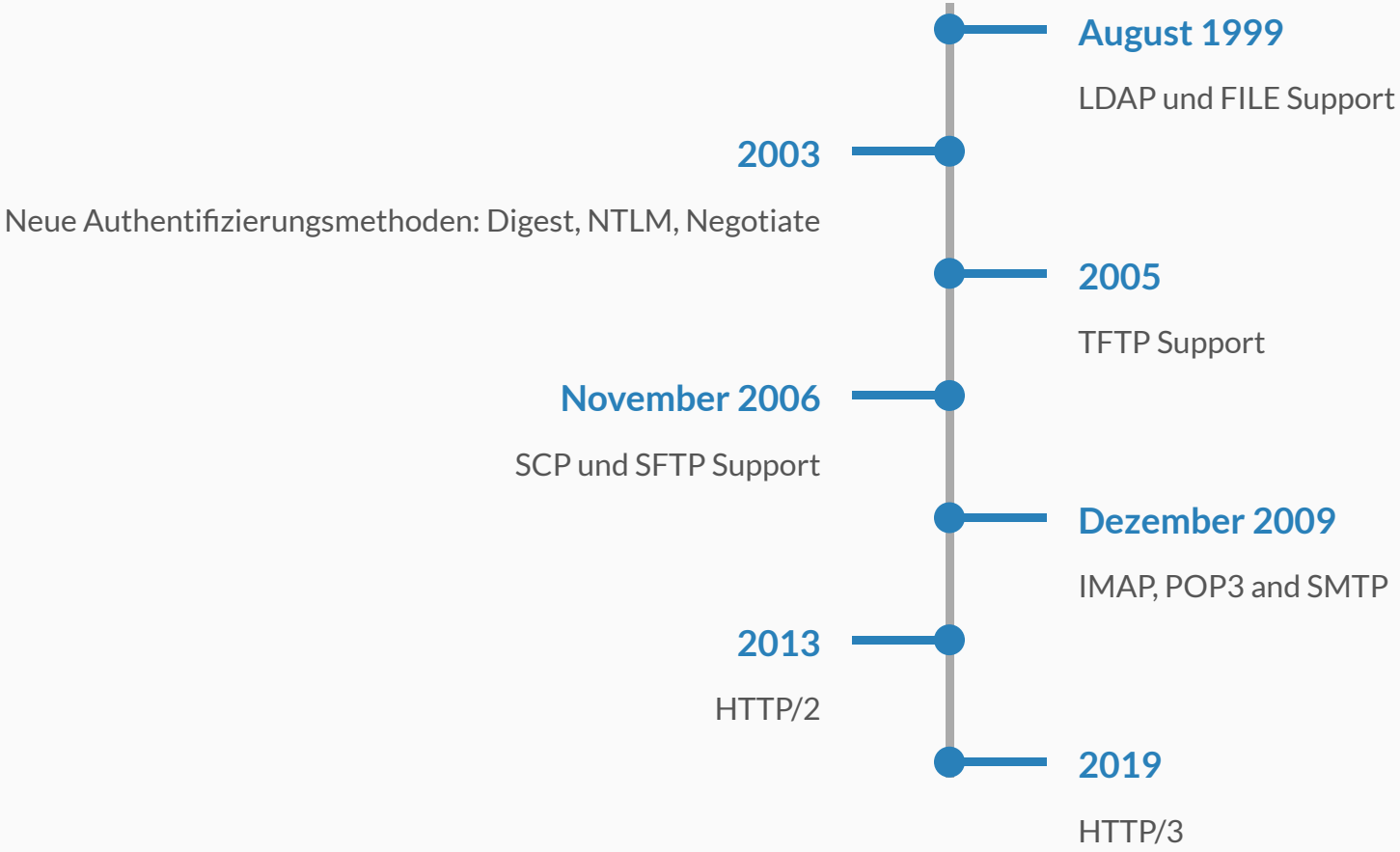


Abbildung 14: „Screenshot GitHub Februar 2026“

Geschichte


11. November 1996
Rafael Sagula veröffentlicht HttpGet version 0.1
17. Dezember 1996
HttpGet 0.2, Daniel Stenberg
Januar 1997
HttpGet 0.3, HTTP Urls in Kommandozeile
8. April 1997
HttpGet 1.0, HTTP Proxy Support
August 1997
urlget 2.0, FTP Support
24. November 1997
urlget 3.1, FTP Upload Support
20. März 1998
curl 4.0

Geschichte


August 1999
LDAP und FILE Support
2003
Neue Authentifizierungsmethoden: Digest, NTLM, Negotiate
2005
TFTP Support
November 2006
SCP und SFTP Support
Dezember 2009
IMAP, POP3 and SMTP
2013
HTTP/2
2019
HTTP/3

API

HTTP Request Methode -X

- mit -X kann man die HTTP Request Methode angeben

HTTP Request `OPTIONS`

- Antwortet mit erlaubten Methoden
- `Allow` erlaubte Request Methoden
- `Access-Control-Allow-Methods` erlaubte CORS Methoden

```bash
curl -X OPTIONS https://other.example.com/homepage -vL
...
< Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE, PUT
< Allow: GET,HEAD,POST,PUT,PATCH,DELETE,OPTIONS
...
```

HTTP Request `OPTIONS`

- `-X HEAD` liefert Metadaten einer Ressource in Form von Headers

Warning: Setting custom HTTP method to HEAD with -X/--request may not work the way you want. Consider using -I/--head instead.

- simuliert `GET` aber liefert keinen Body
- zum Testen um grosse Downloads zu vermeiden

```bash
curl --head https://other.example.com/homepage -vL
...
< Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE, PUT
Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE, PUT
...
```