

NAME

akfweb-dl – Ladeprogramm für HTTP

SYNTAX

akfweb-dl [*Optionen*] [*Datei* ...] <*URL*>

BESCHREIBUNG

Dies ist ein einfaches Ladeprogramm für HTTP.

Es kann z.B. als Backend von anderen Programmen aus aufgerufen werden, um Informationen oder anderes Material aus dem Internet nachzuladen.

Das Programm ist zweisprachig: Deutsch und Englisch.

OPTIONEN

-h

--help

--Hilfe Zeigt eine kurze Hilfe an.

-V

--version

--Version

Zeigt die Version an.

-i

--info Nur Infos über die Adresse auf der Standard-Fehlerausgabe ausgeben. Dies geschieht über eine *HEAD*-Anfrage.

-I

--head Wie --info, aber HTTP-Kopf ungefiltert ausgeben.

-O

--Optionen

--options

Wie --head, aber mit der Methode *OPTIONS*.

-c

--stdout

Den Inhalt über die Standardausgabe ausgeben, anstatt sie in eine Datei zu speichern. Sonstige Ausgaben erfolgen über die Fehlerausgabe.

-o "*Dateiname*"

--output="*Dateiname*"

--Ausgabe="*Dateiname*"

--Datei="*Dateiname*"

Dies als *Dateiname* verwenden. Wenn bereits eine Datei mit diesem *Dateinamen* besteht, bricht das Programm mit einer Fehlermeldung ab. Wird als *Dateiname* nur ein Minus-Zeichen (-) angegeben, erfolgt die Ausgabe über die Standardausgabe, wie bei -c.

-a "*Dateiname*"

--aktualisiere="*Dateiname*"

Aktualisiere *Dateiname*. Wenn die Datei bereits besteht, wird die Änderungszeit an den Server übermittelt mit der Bitte die Datei nur zu übertragen, wenn sie auf dem Server neuer ist. In dem Fall wird die Datei überschrieben. (Server, die diese Funktion nicht vollständig unterstützen, werden die Datei immer übertragen.)

-S

Neu angelegte Dateien sollen nur für den Benutzer zugänglich sein, falls möglich. (Das entspricht »umask 0077«) Das sollte man verwenden, wenn man z. B. temporäre Dateien in /tmp/ anlegt.

Achtung: Das funktioniert aber nicht mit allen Betriebssystemen und Dateisystemen.

--gzip

Akzeptiere komprimierte Dateien. Es gibt aber keine Garantie, dass der Server das komprimiert schickt. Der Dateiname wird angepasst, indem ein .gz oder .Z angefügt wird. Das Programm gzip kann beides entkomprimieren.

-L
--entfernen
--delete Bitte den Server darum, die angegebene Adresse unzugänglich zu machen. In den meisten Fällen wird der Server eine solche Bitte jedoch nur mit einer höhnischen Fehlermeldung quittieren.

-u
--cp
--senden
--upload
 Versuche die angegebenen Dateien an die Adresse zu senden, falls der Server das erlaubt. Wenn bereits entsprechende Dateien auf dem Server liegen und älter sind, werden diese ersetzt. Die Zieladresse wird am Schluss angegeben, wie bei *cp*.
 Die angegebene Adresse sollte mit einem Schrägstrich enden. Dann werden die Dateinamen (ohne Pfad) angehängt. Wenn nur eine einzelne Datei gesendet werden soll, darf die Adresse mit einem neu zu verwendenden Dateinamen enden.

-s
--still nur das Nötigste ausgeben

-l
--links
--Links Gibt eine Liste der Links aus HTML-Code auf der Standardausgabe aus. Das ist vor allem bei Adressen von Verzeichnissen nützlich.

-U
--nicht-umleiten
 Umleitungen als Fehler behandeln

-M
--nicht-interaktiv
 nicht interaktiv nach Zugangsdaten fragen

-N
--nohup
 Das Signal HUP ignorieren. Die Optionen **-s** und **-M** werden implizit mit gesetzt. Dies ist nützlich für längere Downloads auf einem entfernten System, wenn man das Programm im Hintergrund startet und sich danach ausloggen will.

-Z "Name:Kennung"
--Zugang="Name:Kennung"
 Zugangsdaten mitschicken. Die Zugangsdaten können im Klartext oder Base64-kodiert angegeben werden. Diese Option sollte mit Vorsicht verwendet werden. Es gibt bessere Möglichkeiten Zugangsdaten anzugeben.

-P http://[Name:Kennung@]Host:Port/
--proxy=http://[Name:Kennung@]Host:Port/
--Proxy=http://[Name:Kennung@]Host:Port/
 Proxy verwenden. Unterstützte Protokolle: http, socks4, socks4a, socks5, socks5h. Das kann auch über Umgebungsvariablen angegeben werden. Mit **--proxy=""** kann man verhindern, dass ein Proxy verwendet wird.

-t
--tor Tor verwenden. Das ist eine Kurzform von **--Proxy=socks4a://127.0.0.1:9050/**. Es funktioniert nur wenn Tor installiert ist, jedoch nicht mit dem Tor-Browser.

--socket=Socketpfad
--Socket=Socketpfad
 Verbindung über lokalen Socket herstellen. Das kann ein Server oder ein HTTP-Proxy sein.

-r *URL*
--referrer=*URL*
 verweisende *URL* angeben

-A "Name"
--user-agent="Name"
 Verwende *Name* als Angabe für den User-Agent. Diese Angabe ist zum Beispiel für Statistiken interessant. Das kann auch über eine Umgebungsvariable angegeben werden.

-m *E-Mail-Adresse*
--mail=*E-Mail-Adresse*
 Kontakt-Adresse mitschicken. Diese Option sollte man verwenden, wenn man einen Bot schreibt. Über diese Adresse kann man sich dann über das Teil beschweren. Das kann auch über eine Umgebungsvariable angegeben werden.

-T
--trace Gibt auf der Standardausgabe aus, wie die Anfrage beim Server angekommen ist. Damit sieht man zum Beispiel auch Angaben über welche Proxies das lief. Das wird leider nicht von jedem Server unterstützt.

-4
--IPv4
-6
--IPv6 Auf IPv4 oder IPv6 einschränken. Ansonsten wird beides durchprobiert. Bei Verwendung eines Proxys bezieht sich das nur auf die Verbindung zu diesem.

UMGEBUNGSVARIABLEN

all_proxy
ALL_PROXY
http_proxy
 Proxy verwenden.

Die Angabe erfolgt in Form einer URL,
 zum Beispiel: *http://127.0.0.1:8080/*,
 oder mit Zugangsdaten: *http://Name:Kennung@127.0.0.1:8080/*.

Es wird auch SOCKS 4, 4a und 5h unterstützt.
 Zum Beispiel: *socks5h://Name:Kennung@127.0.0.1:1080/*.

Dieses Programm behandelt derzeit socks5 als Alias für socks5h. Zugangsdaten werden unverschlüsselt übertragen. SOCKS 4 und 4a unterstützen gar keine Zugangsdaten, aber da wird die Nutzerkennung übertragen (LOGNAME).

Manche Server können ausgetrickst werden, dass sie bei CGI oder PHP die Umgebungsvariable **HTTP_PROXY** erzeugen. Darum wird sie hier nicht großgeschrieben erlaubt.

HTTPSOCKET

Verbindung über einen lokalen Unix-Socket mit dem angegebenen Pfad herstellen. Das kann ein Server oder ein HTTP-Proxy sein.

AKFUSERAGENT

Angabe für den User-Agent

AKFAGENTMAIL

Kontakt-Adresse mitschicken

LC_ALL, LC_MESSAGES, LANG

anhand dieser Variablen wird entschieden, ob die Ausgabe auf deutsch erfolgt.

HOME Heimatverzeichnis des Benutzers. Hier wird die Datei *.netrc* gesucht.

DATEIEN

`~/.netrc` Zugangsdaten für verschiedene Server. Die Datei darf nur für den Benutzer lesbar sein. Siehe [netrc\(5\)](#).

NOTIZEN

Der Dateiname wird normalerweise aus der URL hergeleitet. Falls die URL auf / endet, wird die Datei »index.html« genannt. Bestehende Dateien werden nicht überschrieben.

Achtung: Falls eine Umleitung erfolgt, ist der automatisch gewählte Dateiname nicht unbedingt der, den man erwartet. In Skripten ist es somit empfehlenswert, den Dateinamen mit der Option `-o` anzugeben, oder die Standardausgabe zu verwenden.

Internationale Domainnamen (IDN oder »Umlaut-Domains«) werden leider nicht unterstützt. Stattdessen muss man die ASCII-Darstellung (ACE-Form, A-Label) verwenden. Zum Beispiel statt »<http://akföster.de/>« muss man »<http://xn--akfrster-p4a.de/>« verwenden.

HTTPS, dh. verschlüsselte Verbindungen werden nicht unterstützt; Webseiten, die HTTPS erzwingen, somit ebenfalls nicht.

Für die Autorisierung wird nur die Methode **Basic** unterstützt. Die Zugangsdaten können über die Datei `~/.netrc` angegeben werden. Ansonsten werden sie interaktiv abgefragt, wenn nicht die Option `-M` oder `-Z` verwendet wurde. Zugangsdaten für einen Proxy werden jedoch nicht interaktiv abgefragt. **Achtung:** Diese Zugangsdaten werden unverschlüsselt übertragen.

Wichtig: Falls man automatisiert mehrere Dateien von einem öffentlichen Server laden will, sollte man eine Wartezeit von vielleicht zwei Sekunden zwischen den Downloads einrichten, um den Server nicht zu stark zu belasten! Denkt daran, dass ihr wohl nicht die einzigen sein werdet, die auf den Server zugreifen; und auch nicht die einzigen, die das automatisiert tun. Nach Möglichkeit sollte man auch die Angaben der Datei `/robots.txt` von dem Server beachten.

Bei jeder Anfrage erfolgt auch eine DNS-Anfrage. Es wird dringend empfohlen einen DNS-Cache-Server zu benutzen. (In Routern ist sowas aber oft schon eingebaut.)

Bei Verwendung eines HTTP-Proxies oder bei SOCKS 4a und 5h erfolgt die Namensauflösung über den Proxy. (Bei diesem Programm ist `socks5` ein Alias für `socks5h`.)

Domains, die auf ».onion« enden, werden nie per DNS abgefragt. Bei Verwendung von Tor sind sie aber benutzbar.

BEKANNTE PROBLEME

Bei der Datei `.netrc` ist es prinzipiell nicht möglich Namen oder Passwörter mit Leerzeichen oder Tabs zu verwenden. Das Format ist aber altbekannt und wird auch von anderer Software verwendet. Darum will ich daran nichts ändern.

AUTOREN

Copyright © 2015-2022 Andreas K. Förster

Dieses Programm ist freie Software. Sie können es unter den Bedingungen der GNU General Public License, wie von der Free Software Foundation veröffentlicht, weitergeben und/oder modifizieren, entweder gemäß Version 3 der Lizenz oder (nach Ihrer Option) jeder späteren Version.

Die Veröffentlichung dieses Programms erfolgt in der Hoffnung, dass es Ihnen von Nutzen sein wird, aber **ohne irgendeine Garantie**, sogar ohne die implizite Garantie der **Marktreife** oder der **Verwendbarkeit für einen bestimmten Zweck**. Details finden Sie in der GNU General Public License.

Sie sollten ein Exemplar der GNU General Public License zusammen mit diesem Programm erhalten haben. Falls nicht, siehe <<http://www.gnu.org/licenses/>>.

SIEHE AUCH

[netrc\(5\)](#), [akfwebserver\(1\)](#) [akfgopher-dl\(1\)](#), [curl\(1\)](#), [wget\(1\)](#),

<https://akfoerster.de/p/akfnetz/>