



昵称： 苍穹冰尘
园龄： 4年6个月
粉丝： 23
关注： 22
+加关注

<	2019年11月						>
日	一	二	三	四	五	六	
27	28	29	30	31	1	2	
3	4	5	6	7	8	9	
10	11	12	13	14	15	16	
17	18	19	20	21	22	23	
24	25	26	27	28	29	30	
1	2	3	4	5	6	7	

搜索

找找看

谷歌搜索

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

积分与排名

积分 - 61133

排名 - 10755

随笔分类 (101)

IDE工具(2)

IT笔试面试(9)

IT杂识(5)

JAVAAE(9)

JAVASE(11)

JAVA并发与网络编程(2)

JAVA分布式(11)

JVM(5)

Linux(16)

计算机基础(3)

计算机网络(7)

生活杂忆(6)

数据库(5)

算法与数据结构(10)

大牛专栏

lanceyan 严澜 (个人)

兰亭风月的专栏

沈涛博客(新浪)

刘正仅博客

stamen的程序员之路 (iteye)

黄赞博客 (开源中国)

Maven仓库管理-Nexus

左耳朵耗子

经典专栏

阿里中间件博客

Li昊林的博客

JOB BLOG (伯乐在线)

最代码

curl命令大全

curl 命令行工具的使用及命令参数说明



curl是一个开源的用于数据传输的命令工具与库，它使用url语法格式，支持众多传输协议，包括：HTTP、HTTPS、FTP、FTPS、GOPHER、TFTP、SCP、SFTP、SMB、TELNET、DICT、LDAP、LDAPS、FILE、IMAP、SMTP、POP3、RTSP和RTMP。curl库提供了很多强大的功能，你可以利用它来进行HTTP/HTTPS请求、上传/下载文件等，且支持Cookie、认证、代理、限速等。

1. curl的使用
 - 1.1 URL访问
 - 1.2 表单提交
 - 1.3 其它HTTP请求方法
 - 1.4 文件上传
 - 1.5 HTTPS支持
 - 1.6 添加请求头
 - 1.7 Cookie支持
2. curl语法及选项

1. curl的使用

1.1 URL访问

访问一个网页时，可以使用curl命令后加上要访问的网址：

```
$ curl itbilu.com
<html>
<head><title>301 Moved Permanently</title></head>
<body bgcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<hr/><center>nginx/1.6.2</center>
</body>
</html>
```

如上所示，我们就看到所访问网址的页面源码。

重定向跟踪

在上图示例中，页面使用了301重定向，这时我们可以添加-l参数来跟踪URL重定向：

```
$ curl -L itbilu.com
```

页面保存

如果需要将页面源码保存到本地，可以使用-o参数：

```
$ curl -o [文件名] itbilu.com
```

查看头信息

如果需要查看访问页面的可以添加-i或--include参数：

```
$ curl -i itbilu.com
```

添加-i参数后，页面响应头会和页面源码（响应体）一块返回。如果只想查看响应头，可以使用-I或--head参数：

```
$ curl -I itbilu.com
HTTP/1.1 301 Moved Permanently
Server: nginx/1.6.2
Date: Sun, 25 Jun 2017 02:03:45 GMT
Content-Type: text/html
Content-Length: 184
Connection: keep-alive
Location: https://itbilu.com/
```

1.2 表单提交

通过Form表单，可以将Web页面的表单数据提交到服务端。提交表单时，可以使用GET或POST提交方法。

curl同样支持表单数据提交，也可以使用GET或POST提交方法。

GET数据提交

当全用GET表单数据提交时，提交数据会被附加到请求URL的后面，类型如下：

```
$ curl 'https://itbilu.com/?keyword=linux&page=3'
```

使用curl进行GET数据提交时，也可以直接把提交数据添加到URL后面：

```
curl https://itbilu.com/?keyword=linux&page=3
```

POST数据提交

curl使用POST提交表单数据时，除了-X参数指定请求方法外，还要使用--data参数添加提交数据：

```
$ curl -X POST --data 'keyword=linux' itbilu.com
```

1.3 其它HTTP请求方法

目前为止，我们使用GET和POST两种HTTP请求。curl支持所有HTTP请求方法，只要通过-X参数指定即可。

如，使用DELETE请求：

```
$ curl -X DELETE itbilu.com/example.html
```

使用PUT请求，并指定请求数据：

```
$ curl -X PUT --data 'keyword=linux' itbilu.com
```

1.4 文件上传

curl支持文件上传，上传文件时使用-T或--upload-file参数：

```
$ curl -T ./index.html www.uploadhttp.com/receive.cgi
```

1.5 HTTPS支持

对于使用了SSL/TLS加密的HTTPS协议，可以使用curl直接访问：

```
$ curl https://itbilu.com
```

如果使用的本地ssl证书认证的HTTPS，可以通过-k或--cert参数指定本地证书：

校友职场

蜗牛的专栏

heinz_ho的专栏

天亮博客

陆神博客

阅读排行榜

1. Java的Synchronized, 对象锁和类锁举例(15494)

2. QPS的计算方法(6040)

3. linux中cat, more, less, tail, head命令的区别(4479)

4. 几种MQ消息队列对比与消息队列之间的通信问题(4466)

5. linux服务端load的含义(3867)

```
$ curl -E nycert.pem https://itbilu.com
```

1.6 添加请求头

有时在进行HTTP请求时，需要自定义请求头。在curl中，可以通过-H或—header参数来指定请求头。多次使用-H或—header参数可指定多个请求头。

如，指定Content-Type及Authorization请求头：

```
$ curl -H 'Content-Type:application/json' -H 'Authorization: bearer eyJhbGciOiJIUzI1NiJ9' itbilu.com
```

1.7 Cookie支持

Cookie是一种常用的保持服务会话信息的方法，curl也支持使用Cookie。

可以通过—cookie参数指定发送请求时的Cookie值，也可以通过-b [文件名]来指定一个存储了Cookie值的本地文件：

```
$ curl -b stored_cookies_in_file itbilu.com
```

Cookie值可能会被服务器所返回的值所修改，并应用于下次HTTP请求。这时，可以能通过-e参数指定存储服务器返回Cookie值的存储文件：

```
$ curl -b cookies.txt -o newcookies.txt itbilu.com
```

除以上用法外，curl还可以设置用户代理（客户端）信息、使用代理服务器、指定认证用户名 / 密码等。详见：[curl语法及选项](#)

```
curl --silent -H "Host: www.test.com" "192.168.0.1/xxx/xxx/t.php"
curl "http://www.test.com/LiveUserCount.ac" -x 127.0.0.1:1080
```

2. curl语法及选项

语法

curl语法结构如下：

```
curl [options...] <url>
```

参数选项

curl (7.29.0) 所支持的选项 (options) 参数如下：

在以下选项中，(H) 表示仅适用 HTTP/HTTPS，(F) 表示仅适用于 FTP

- anyauth 选择 “any” 认证方法 (H)
- a, —append 添加要上传的文件 (F/SFTP)
- basic 使用HTTP基础认证 (Basic Authentication) (H)
- cacert FILE CA 证书，用于每次请求认证 (SSL)
- capath DIR CA 证书目录 (SSL)
- E, —cert CERT[:PASSWD] 客户端证书文件及密码 (SSL)
- cert-type TYPE 证书文件类型 (DER/PEM/ENG) (SSL)
- ciphers LIST SSL 秘钥 (SSL)
- compressed 请求压缩 (使用 deflate 或 gzip)
- K, —config FILE 指定配置文件
- connect-timeout SECONDS 连接超时设置
- C, —continue-at OFFSET 断点续传
- b, —cookie STRING/FILE Cookies字符串或读取Cookies的文件位置 (H)
- c, —cookie-jar FILE 操作结束后，要写入 Cookies 的文件位置 (H)
- create-dirs 创建必要的本地目录层次结构
- curlf 在上传时将 LF 转写为 CRLF
- curlfile FILE 从指定的文件获得FPM格式CRL列表
- d, —data DATA HTTP POST 数据 (H)
- data-ascii DATA ASCII 编码 HTTP POST 数据 (H)
- data-binary DATA binary 编码 HTTP POST 数据 (H)
- data-urlencode DATA url 编码 HTTP POST 数据 (H)
- delegation STRING GSS-API 委托权限
- digest 使用数字身份验证 (H)
- disable-eprt 禁止使用 EPRT 或 LPRT (F)
- disable-epsv 禁止使用 EPSV (F)
- D, —dump-header FILE 将头信息写入指定的文件
- egd-file FILE 为随机数设置EGD socket路径 (SSL)
- engine ENGINE 加密引擎 (SSL)。 “engine list” 指定列表
- f, —fail 连接失败时不显示HTTP错误信息 (H)
- F, —form CONTENT 模拟 HTTP 表单数据提交 (multipart POST) (H)
- form-string STRING 模拟 HTTP 表单数据提交 (H)
- ftp-account DATA 帐户数据提交 (F)
- ftp-alternative-to-user COMMAND 指定替换 “USER [name]” 的字符串 (F)
- ftp-create-dirs 如果不存在则创建远程目录 (F)
- ftp-method [MULTICWD/NOCWD/SINGLECWD] 控制 CWD (F)
- ftp-pasv 使用 PASV/EPSV 替换 PORT (F)
- P, —ftp-port ADR 使用指定 PORT 及地址替换 PASV (F)
- ftp-skip-pasv-ip 跳过 PASV 的IP地址 (F)
- ftp-pret 在 PASV 之前发送 PRET (drftpd) (F)
- ftp-ssl-ccc 在认证之后发送 CCC (F)
- ftp-ssl-ccc-mode ACTIVE/PASSIVE 设置 CCC 模式 (F)
- ftp-ssl-control ftp 登录时需要 SSL/TLS (F)
- G, —get 使用 HTTP GET 方法发送 -d 数据 (H)
- globoff 禁用的 URL 队列 及范围使用 {} 和 []
- H, —header LINE 要发送到服务端的自定义请求头 (H)
- I, —head 仅显示响应文档头
- h, —help 显示帮助
- O, —http1.0 使用 HTTP 1.0 (H)
- ignore-content-length 忽略 HTTP Content-Length 头
- i, —include 在输出中包含协议头 (H/F)
- k, —insecure 允许连接到 SSL 站点，而不使用证书 (H)
- interface INTERFACE 指定网络接口 / 地址
- 4, —ipv4 将域名解析为 IPv4 地址
- 6, —ipv6 将域名解析为 IPv6 地址
- j, —junk-session-cookies 读取文件中但忽略会话cookie (H)
- keepalive-time SECONDS keepalive 时间间隔
- key KEY 私钥文件名 (SSL/SSH)
- key-type TYPE 私钥文件类型 (DER/PEM/ENG) (SSL)
- krb LEVEL 启用指定安全级别的 Kerberos (F)
- libcurl FILE 命令的libcurl等价代码
- limit-rate RATE 限制传输速度
- l, —list-only 只列出FTP目录的名称 (F)
- local-port RANGE 强制使用的本地端口号
- L, —location 跟踪重定向 (H)
- location-trusted 类似 —location 并发送验证信息到其它主机 (H)
- M, —manual 显示全手册
- mail-from FROM 从这个地址发送邮件
- mail-rcpt TO 发送到到这个接收人(s)
- mail-auth AUTH 原始电子邮件的起始地址
- max-filesize BYTES 下载的最大文件大小 (H/F)
- max-redirs NUM 最大重定向数 (H)
- m, —max-time SECONDS 允许的最多传输时间
- metalink 处理指定的URL上的XML文件
- negotiate 使用 HTTP Negotiate 认证 (H)
- n, —netrc 必须从 .netrc 文件读取用户名和密码
- netrc-optional 使用 .netrc 或 URL: 将重写 -n 参数
- netrc-file FILE 设置要使用的 netrc 文件名
- N, —no-buffer 禁用输出流的缓存
- no-keepalive 禁用 connection 的 keepalive
- no-sessionid 禁止重复使用 SSL session-ID (SSL)
- no-proxy 不使用代理的主机列表
- ntlm 使用 HTTP NTLM 认证 (H)
- o, —output FILE 将输出写入文件，而非 stdout

```
--pass PASS      传递给私钥的短语 (SSL/SSH)
--post301      在 301 重定向后不要切换为 GET 请求 (H)
--post302      在 302 重定向后不要切换为 GET 请求 (H)
--post303      在 303 重定向后不要切换为 GET 请求 (H)
-#, --progress-bar 以进度条显示传输进度
--proto PROTOCOLS 启用/禁用 指定的协议
--proto-redir PROTOCOLS 在重定向上 启用/禁用 指定的协议
-z, --proxy [PROTOCOL://]HOST[:PORT] 在指定的端口上使用代理
    --proxy-anyauth 在代理上使用 "any" 认证方法 (H)
    --proxy-basic 在代理上使用 Basic 认证 (H)
    --proxy-digest 在代理上使用 Digest 认证 (H)
    --proxy-negotiate 在代理上使用 Negotiate 认证 (H)
    --proxy-ntlm 在代理上使用 NTLM 认证 (H)
-U, --proxy-user USER[:PASSWORD] 代理用户名及密码
    --proxy1.0 HOST[:PORT] 在指定的端口上使用 HTTP/1.0 代理
-p, --proxytunnel 使用HTTP代理 (用于 CONNECT)
-pubkey KEY      公钥文件名 (SSH)
-Q, --quote CMD    在传输开始前向服务器发送命令 (F/SFTP)
--random-file FILE 读取随机数据的文件 (SSL)
-r, --range RANGE  仅检索范围内的字节
--raw            使用原始HTTP传输, 而不使用编码 (H)
-e, --referer      Referer URL (H)
-J, --remote-header-name 从远程文件读取头信息 (H)
-O, --remote-name  将输出写入远程文件
    --remote-name-all 使用所有URL的远程文件名
-R, --remote-time  将远程文件的时间设置在本地输出上
-X, --request COMMAND 使用指定的请求命令
    --resolve HOST:PORT:ADDRESS 将 HOST:PORT 强制解析到 ADDRESS
    --retry NUM      出现问题时的重试次数
    --retry-delay SECONDS 重试时的延时时长
    --retry-max-time SECONDS 仅在指定时间段内重试
-S, --show-error  显示错误. 在选项 -s 中, 当 curl 出现错误时将显示
-s, --silent      Silent模式. 不输出任务内容
    --socks4 HOST[:PORT] 在指定的 host + port 上使用 SOCKS4 代理
    --socks4a HOST[:PORT] 在指定的 host + port 上使用 SOCKS4a 代理
    --socks5 HOST[:PORT] 在指定的 host + port 上使用 SOCKS5 代理
    --socks5-hostname HOST[:PORT] SOCKS5 代理, 指定用户名、密码
    --socks5-gssapi-service NAME 为gssapi使用SOCKS5代理服务名称
    --socks5-gssapi-mec 与MEC SOCKS5服务器兼容
-Y, --speed-limit RATE 在指定限速时间之后停止传输
-y, --speed-time SECONDS 指定时间之后触发限速. 默认 30
--ssl          尝试 SSL/TLS (FTP, IMAP, POP3, SMTP)
--ssl-reqd     需要 SSL/TLS (FTP, IMAP, POP3, SMTP)
-2, --sslv2    使用 SSLv2 (SSL)
-3, --sslv3    使用 SSLv3 (SSL)
    --ssl-allow-beast 允许的安全漏洞, 提高互操作性 (SSL)
--stderr FILE  重定向 stderr 的文件位置. - means stdout
--tcp-nodelay  使用 TCP_NODELAY 选项
-t, --telnet-option OPT=VAL 设置 telnet 选项
    --tftp-blksize VALUE 设备 TFTP BLKSIZE 选项 (必须 >512)
-z, --time-cond TIME 基于时间条件的传输
-1, --tlsv1     使用 => TLSv1 (SSL)
    --tlsv1.0  使用 TLSv1.0 (SSL)
    --tlsv1.1  使用 TLSv1.1 (SSL)
    --tlsv1.2  使用 TLSv1.2 (SSL)
--trace FILE   将 debug 信息写入指定的文件
--trace-ascii FILE 类似 --trace 但使用16进制输出
--trace-time   向 trace/verbose 输出添加时间戳
--tr-encoding  请求压缩传输编码 (H)
-T, --upload-file FILE 将文件传输 (上传) 到指定位置
    --url URL    指定所使用的 URL
-B, --use-ascii 使用 ASCII/text 传输
-u, --user USER[:PASSWORD] 指定服务器认证用户名、密码
    --tlshuser USER TLS 用户名
    --tlspassword STRING TLS 密码
    --tlsvauthype STRING TLS 认证类型 (默认 SRP)
--unix-socket FILE 通过这个 UNIX socket 域连接
-A, --user-agent STRING 要发送到服务器的 User-Agent (H)
-v, --verbose    显示详细操作信息
-V, --version    显示版本号并退出
-v, --write-out FORMAT 完成后输出什么
    --zattr      将元数据存储在扩展文件属性中
-q             .curlrc 如果作为第一个参数无效
```

转载: https://itbll.com/linux/man/4y29qH_7X.html

分类: [Linux](#)




 苍穹冰尘
关注 - 22
粉丝 - 23
[+加关注](#)

1 0
[推荐](#) [反对](#)

« 上一篇: 笔记整理
» 下一篇: 内存优化以及区别

posted @ 2018-12-26 17:36 苍穹冰尘 阅读(2322) 评论(0) 编辑 收藏

 注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问 网站首页](#)。

【推荐】超50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【培训】马士兵老师强势回归! Java线下课程全免费, 双十一大促!

【活动】京东云服务器_云主机低于1折, 低价高性能产品备战双11

【推荐】天翼云双十一翼降到底, 云主机11.11元起, 抽奖送大礼

【优惠】腾讯云 11.11智慧上云, 爆款提前购与双11活动同价

【福利】个推四大热门移动开发SDK全部免费用一年, 限时抢!

【优惠】七牛云采购嘉年华, 云存储、CDN等云产品低至1折

相关博文:

- curl命令
- curl命令
- curl命令的基本使用
- curl命令使用
- curl命令备注
- » 更多推荐...