

## Linux 中 CURL常用命令详解

下载单个文件，默认将输出打印到标准输出中(STDOUT)中

通过-o/-O选项保存下载的文件到指定的文件中：

同时获取多个文件

断点续传

对CURL使用网络限速

CURL授权

从FTP服务器下载文件

获取更多信息

为CURL设置代理

保存与使用网站cookie信息

传递请求数据

# GET

# POST

除了使用GET和POST协议外，还可以通过 -X 选项指定其它协议，如：

## 介绍：

cURL是一个利用URL语法在命令行下工作的文件传输工具，1997年首次发行。它支持文件上传和下载，所以是综合传输工具，但按传统，习惯称cURL为下载工具。cURL还包含了用于程序开发的libcurl。

```
1 发送get:
2  curl localhost:5000
3  form:
4  curl -d a=a localhost:5000
5  json:
6  curl -d '{"num":1}' -H 'Content-Type:application/json' localhost:5000
7
8  详细文档:
9  https://www.jianshu.com/p/07c4dddae43a
```

## Linux 中 CURL常用命令详解

## 下载单个文件，默认将输出打印到标准输出中(STDOUT)中

```
1 curl http://www.centos.org
```

### 通过-o/-O选项保存下载的文件到指定的文件中：

-o：将文件保存为命令行中指定的文件名的文件中

-O：使用URL中默认的文件名保存文件到本地

```
1 # 将文件下载到本地并命名为mygettext.html
2 curl -o mygettext.html
3 http://www.gnu.org/software/gettext/manual/gettext.html
4 # 将文件保存到本地并命名为gettext.html
   curl -O http://www.gnu.org/Software/gettext/manual/gettext.html
```

同样可以使用转向字符">"对输出进行转向输出

### 同时获取多个文件

```
1 curl -O URL1 -O URL2
```

若同时从同一站点下载多个文件时，curl会尝试重用链接(connection)。

通过-L选项进行重定向

默认情况下CURL不会发送HTTP Location headers(重定向).当一个被请求页面移动到另一个站点时，会发送一个HTTP Location

header作为请求，然后将请求重定向到新的地址上。

例如：访问google.com时，会自动将地址重定向到google.com.hk上。

?

```
1 curl http://www.google.com
2 <HTML>
3 <HEAD>
4   <meta http-equiv="content-type" Content="text/html; charset=utf-">
5   <TITLE> Moved</TITLE>
6 </HEAD>
7 <BODY>
8   <H> Moved</H>
9   The document has moved
10  <A HREF="http://www.google.com.hk/url?sa=p&hl=zh-CN&pref=hkredirect&pval=yes&q=http://www.google.com.hk/&ust=amp;usg=AFQjCNFo
11  </BODY>
12 </HTML>
```

上述输出说明所请求的档案被转移到了<http://www.google.com.hk>。

这是可以通过使用-L选项进行强制重定向

1 # 让curl使用地址重定向，此时会查询google.com.hk站点

2 curl -L <http://www.google.com>

## 断点续传

通过使用-C选项可对大文件使用断点续传功能，如：

?

```
1 # 当文件在下载完成之前结束该进程
2 $ curl -O http://www.gnu.org/software/gettext/manual/gettext.html
3 #####. %
4 # 通过添加-C选项继续对该文件进行下载，已经下载过的文件不会被重新下载
5 curl -C - -O http://www.gnu.org/software/gettext/manual/gettext.html
6 #####. %
```

## 对CURL使用网络限速

通过--limit-rate选项对CURL的最大网络使用进行限制

# 下载速度最大不会超过1000B/second

curl --limit-rate 1000B -O <http://www.gnu.org/software/gettext/manual/gettext.html>

下载指定时间内修改过的文件

当下载一个文件时，可对该文件的最后修改日期进行判断，如果该文件在指定日期内修改过，就进行下载，否则不下载。

该功能可通过使用-z选项来实现：

# 若yy.html文件在2011/12/21之后有过更新才会进行下载

curl -z 21-Dec-11 <http://www.example.com/yy.html>

## CURL授权

在访问需要授权的页面时，可通过-u选项提供用户名和密码进行授权

curl -u username:password URL

# 通常的做法是在命令行只输入用户名，之后会提示输入密码，这样可以保证在查看历史记录时不会将密码泄露

curl -u username URL

## 从FTP服务器下载文件

CURL同样支持FTP下载，若在url中指定的是某个文件路径而非具体的某个要下载的文件名，CURL则会列出该目录下的所有文件名而并非下载该目录下的所有文件

# 列出public\_html下的所有文件夹和文件

```
curl -u ftpuser:ftppass -O ftp://ftp_server/public_html/
```

# 下载xss.php文件

```
curl -u ftpuser:ftppass -O ftp://ftp_server/public_html/xss.php
```

上传文件到FTP服务器

通过 -T 选项可将指定的本地文件上传到FTP服务器上



```
1      # 将myfile.txt文件上传到服务器
2
3      curl -u ftpuser:ftppa$ -T myfile.txt ftp://ftp.testserver.com
4
5      # 同时上传多个文件
6
7      curl -u ftpuser:ftppa$ -T "{file1,file2}" ftp://ftp.testserver.com
8
9      # 从标准输入获取内容保存到服务器指定的文件中
10
11     curl -u ftpuser:ftppa$ -T - ftp://ftp.testserver.com/myfile_1.txt
```

## 获取更多信息

通过使用 -v 和 -trace获取更多的链接信息

## 通过字典查询单词

# 查询bash单词的含义

```
curl dict://dict.org/d:bash
```

# 列出所有可用词典

```
curl dict://dict.org/show:db
```

# 在foldoc词典中查询bash单词的含义

```
curl dict://dict.org/d:bash:foldoc
```

## 为CURL设置代理

-x 选项可以为CURL添加代理功能

# 指定代理主机和端口

```
curl -x proxysever.test.com:3128 http://google.co.in
```

## 其他网站整理

## 保存与使用网站cookie信息

# 将网站的cookies信息保存到sugarcookies文件中

```
curl -D sugarcookies http://localhost/sugarcrm/index.php
```

# 使用上次保存的cookie信息

```
curl -b sugarcookies http://localhost/sugarcrm/index.php
```

## 传递请求数据

默认curl使用GET方式请求数据，这种方式下直接通过URL传递数据

可以通过 --data/-d 方式指定使用POST方式传递数据

### # GET

```
curl -u username https://api.github.com/user?access\_token=XXXXXXXXXX
```

### # POST

```
curl -u username --data "param=value¶m=value" https://api.github.com
```

# 也可以指定一个文件，将该文件中的内容当作数据传递给服务器端

```
curl --data @filename https://github.api.com/authorizations
```

注：默认情况下，通过POST方式传递过去的的数据中若有特殊字符，首先需要将特殊字符转义在传递给服务器端，如value值中包含有空格，则需要先将空格转换成%20，如：

```
curl -d "value%201" http://hostname.com
```

在新版本的CURL中，提供了新的选项 --data-urlencode，通过该选项提供的参数会自动转义特殊字符。

```
curl --data-urlencode "value 1" http://hostname.com
```

除了使用GET和POST协议外，还可以通过 -X 选项指定其它协议，如：

```
curl -I -X DELETE https://api.github.com
```

上传文件

```
curl --form "fileupload=@filename.txt" http://hostname/resource
```