# CURL

#### 获取页面内容

|  |
| --- |
| curl http://www.codebelief.com |

默认使用GET请求。

#### 显示HTTP头

只显示HTTP头，不显示文件内容 ，使用参数： **-I**

|  |
| --- |
| curl -I <http://www.codebelief.com> |

同时显示HTPP头和文件内容，使用参数：**-i**

|  |
| --- |
| curl -i http://www.codebelief.com |

#### 保存到文件

将输出重定向到文件：

|  |
| --- |
| curl http://www.codebelief.com **>** index.html |

或使用curl自带的-o/-O

-o: 结果会被保存到命令行中提供的文件名

-O: URL 中的文件名会被用作保存输出的文件名

|  |
| --- |
| curl -o index.html http://www.codebelief.com  curl -O http://www.codebelief.com/page/2 |

注意：使用 -O 选项时，必须确保链接末尾包含文件名，否则 curl 无法正确保存文件。如果遇到链接中无文件名的情况，应该使用 -o 选项手动指定文件名，或使用重定向符号。

#### 同时下载多个文件

|  |
| --- |
| curl -O http://www.codebelief.com/page/2 -O <http://www.codebelief.com/page/3>  curl -o page1.html http://www.codebelief.com/page/1 -o page2.html http://www.codebelief.com/page/2 |

#### 使用 -L 跟随链接重定向

如果直接使用 curl 打开某些被重定向后的链接，这种情况下就无法获取我们想要的网页内容。

|  |
| --- |
| curl <http://codebelief.com>  结果：  <html>  <head><title>301 Moved Permanently</title></head>  <body bgcolor="white">  <center><h1>301 Moved Permanently</h1></center>  <hr><center>nginx/1.10.3</center>  </body>  </html> |

但通过浏览器打开该链接，会自动跳转到<http://www.codebelief.com>，

此时，使用curl -L 来跟踪重定向：

|  |
| --- |
| curl -L <http://codebelief.com> |

#### 使用 -A 自定义 User-Agent

模拟火狐浏览器：

|  |
| --- |
| curl -A "Mozilla/5.0 (Android; Mobile; rv:35.0) Gecko/35.0 Firefox/35.0" http://www.baidu.com |

#### 使用 -H 自定义 header

|  |
| --- |
| curl **-H** "Referer: www.example.com" **-H** "User-Agent: Custom-User-Agent" http://www.baidu.com |

传递cookie

|  |
| --- |
| curl -H "**Cookie**: JSESSIONID=D0112A5063D938586B659EF8F939BE24" http://www.example.com |

#### 使用 -c 保存 Cookie

-c 后面跟文件名，将cookie信息保存到对应的文件中

|  |
| --- |
| curl -c "cookie-example" http://www.example.com |

#### 使用 -b 读取 Cookie

|  |
| --- |
| curl -b "JSESSIONID=D0112A5063D938586B659EF8F939BE24" http://www.example.com |

这里不同与-H ，-b后直接是cookie指，而-H 后是 Cookie: xxx

若要从文件中读取cookie, -H就不行了

|  |
| --- |
| curl -b "cookie-example" http://www.example.com |

所以-b后面既可以是cookie值，也可以是存放cookie的文件名

#### 使用 -d 发送 POST 请求

-d 指定发送数据

-X 指定发送方式

|  |
| --- |
| curl -d "userName=tom&passwd=123456" -X POST http://www.example.com/login |

使用-d时，如果省略-X, 默认使用POST.

|  |
| --- |
| curl -d "userName=tom**&**passwd=123456" http://www.example.com/login |

带cookie登录：

-c 保存cookie

|  |
| --- |
| curl -c "cookie-login" -d "userName=tom&passwd=123456" http://www.example.com/login |

下次访问时：

|  |
| --- |
| curl -b "cookie-login" http://www.example.com/login |

# wget

#### 命令

* wget url

默认以最后一个”/”的后面的字符来做文件名

* wget -O 新文件名 url

指定文件名

* wget -c url 断点续传
* wget -b url 后台下载

tail -f wget-log查看下载进度

* wget -i fileList.txt 下载多个文件

需要将多个url放到fileList.txt文件里