

Use the `-I` or `-head` option with `curl` to dump only the HTTP headers, without downloading the remote file. For example:

```
$ curl -I http://slynux.org
HTTP/1.1 200 OK
Date: Sun, 01 Aug 2010 05:08:09 GMT
Server: Apache/1.3.42 (Unix) mod_gzip/1.3.26.1a mod_log_bytes/1.2
mod_bwlimited/1.4 mod_auth_passthrough/1.8 FrontPage/5.0.2.2635 mod_
ssl/2.8.31 OpenSSL/0.9.7a
Last-Modified: Thu, 19 Jul 2007 09:00:58 GMT
ETag: "17787f3-3bb0-469f284a"
Accept-Ranges: bytes
Content-Length: 15280
Connection: close
Content-Type: text/html
```

## See also

- ▶ The *Posting to a web page and reading response* recipe in this chapter

## Accessing Gmail e-mails from the command line

Gmail is a widely-used free e-mail service from Google—<http://mail.google.com/>. It allows you to read your mail via authenticated RSS feeds. We can parse the RSS feeds with the sender's name, and an e-mail with a subject. It will help us to have a look at the unread e-mails in the inbox, without opening the web browser.

## How to do it...

Let's go through the shell script to parse the RSS feeds for Gmail to display the unread mails:

```
#!/bin/bash
#Desc: Fetch gmail tool

username='PUT_USERNAME_HERE'
password='PUT_PASSWORD_HERE'

SHOW_COUNT=5 # No of recent unread mails to be shown

echo
```

```

curl -u $username:$password --silent "https://mail.google.com/mail/
feed/atom" | \
tr -d '\n' | sed 's:</entry>:\n:g' | \
sed -n 's/.*<title>\(.*\)</title.*<author><name>\([^<]*\)</
name><email>
\([^<]*\)*/From: \2 [\3] \nSubject: \1\n/p' | \
head -n $(( $SHOW_COUNT * 3 ))

```

The output will be as follows:

```

$ ./fetch_gmail.sh
From: SLYNEX [ slynux@slynux.com ]
Subject: Book release - 2

From: SLYNEX [ slynux@slynux.com ]
Subject: Book release - 1
.
... 5 entries

```



If you are using a Google Mail account with two factor authentication, you will have to generate a new key for this script and use it. Your regular password won't work.

## How it works...

The script uses cURL to download the RSS feed by using user authentication. User authentication is provided by the `-u user:pass` argument. You can use `-u user` without providing the password. Then, while executing cURL, it will interactively ask for the password. Here, you can split the piped commands into different blocks to illustrate how they work.

`tr -d '\n'` removes the newline character, so that we restructure each e-mail entry with `\n` as the delimiter.

`sed 's:</entry>:\n:g'` replaces every `</entry>` element with a newline, so that each e-mail entry is delimited by a new line and, hence, mails can be parsed one-by-one. Have a look at the source of <https://mail.google.com/mail/feed/atom> for XML tags used in the RSS feeds. `<entry> TAGS </entry>` corresponds to a single mail entry.