# A primer on cURL

**cURL** is a powerful utility that supports protocols, such as HTTP, HTTPS, and FTP. It supports many features including POST, cookies, authentication, downloading partial files from a specified offset, referer, user agent string, extra headers, limit speed, maximum file size, progress bar, and so on. cURL is very useful when we play around with automating a web page usage sequence, and to retrieve data. This recipe is a list of the most important features of cURL.

## Getting ready

cURL mostly doesn't come with any Linux distros; you may have to install it by using the package manager. By default, distributions ship with wget.

cURL usually dumps the downloaded files to stdout, and progress information to stderr. To avoid the progress information from being shown, we use the --silent option.

## How to do it...

The curl command can be used to perform different activities, such as downloading, sending different HTTP requests, and specifying HTTP headers. Let's see how we can perform different tasks with cURL:

- To dump the downloaded file onto the terminal (the downloaded data is written to stdout, use the following command:

  ```
  $ curl URL
  ```

- To prevent the curl command from displaying the progress information, mention --silent.

  ```
  $ curl URL --silent
  ```

- To write the downloaded data into a file with the filename parsed from the URL rather than writing into the standard output, use the -O option:

  ```
  $ curl URL --silent -O
  ```

- To show the # progress bar while downloading, use --progress instead of --silent:

  ```
  $ curl http://slynux.org -o index.html --progress
  ################################## 100.0%
  ```

## How it works...

cURL writes a web page or file to the filename as in the URL, instead of writing to stdout. If the filenames are not present in the URL, it will produce an error. Hence, make sure that the URL is a URL to a remote file. curl http://slynux.org -O --silent will display an error, since the filename cannot be parsed from the URL. In such cases, we can manually specify the filename as follows, by using the -o option:

```
$ curl URL --silent -o new_filename
```

## There's more...

In the preceding sections, we learned how to download files and dump HTML pages to the terminal. There are several advanced options that come along with cURL. Let us explore more of cURL.

### Continuing and resuming downloads

cURL has advanced resume download features to continue at a given offset. It helps to download portions of files by specifying an offset:

```
$ curl URL/file -C offset
```

The offset is an integer value in bytes.

cURL doesn't require us to know the exact byte offset, if we want to resume downloading a file. If you want cURL to figure out the correct resume point, use the -C - option, as follows:

```
$ curl -C - URL
```

cURL will automatically figure out where to restart the download of the specified file.

### Setting the referer string with cURL

**Referer** is a string in the HTTP header, used to identify the page from which the user reaches the current web page. When a user clicks on a link on web page A to go to web page B, the referer header string for page B will contain the URL of page A.

Some dynamic pages check the referer string before returning the HTML data. For example, a web page shows a Google logo attached page when a user navigates to a website by searching on Google, and shows a different page when the user navigates to the web page by manually typing the URL.

The author of a website can write a condition to return a Google page if the referer is www.google.com, or else return a different page.