

Posting to a web page and reading the response

POST and GET are two types of requests in HTTP to send information to, or retrieve information from a website. In a GET request, we send parameters (name-value pairs) through the webpage URL itself. In the case of POST, it won't be attached with the URL. POST is used when a form needs to be submitted. For example, a username, the password to be submitted, and the login page to be retrieved.

Posting to pages is used frequently while writing scripts based on webpage retrievals. Let us see how to work with POST. Automating the HTTP GET and POST request by sending the POST data and retrieving the output is a very important task that we practice while writing shell scripts that parse data from websites.

Getting ready

For this recipe, we will use a test website (<http://book.sarathlakshman.com/lsc/mlogs/>), which is used to submit the current user information, such as the hostname and username. In the home page of the website, there are two fields **HOSTNAME** and **USER**, and a **SUBMIT** button. When the user enters a hostname, a user name, and clicks on the **Submit** button, the details will be stored in the website. This process can be automated by using a single line of `curl` (or `wget`) command by automating the POST request. Let's see how to do this by using `curl` or `wget`.

How to do it...

POST and read the HTML response from a real website using `curl` as follows:

```
$ curl URL -d "postvar=postdata2&postvar2=postdata2"
```

For example,

```
$ curl http://book.sarathlakshman.com/lsc/mlogs/submit.php -d "host=test-host&user=slynux"
<html>
You have entered :
<p>HOST : test-host</p>
<p>USER : slynux</p>
<html>
```

`curl` prints the response page.

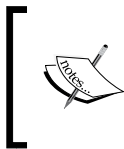
`-d` is the argument used for posting. The string argument for `-d` is similar to the GET request semantics. `var=value` pairs are to be delimited by `&`.

You can POST the data using `wget` by using `--post-data "string"`. For example:

```
$ get http://book.sarathlakshman.com/lsc/mlogs/submit.php --post-data
"host=test-host&user=slynux" -O output.html

$ cat output.html
<html>
You have entered :
<p>HOST : test-host</p>
<p>USER : slynux</p>
<html>
```

Use the same format as `cURL` for name-value pairs.



The string to the post arguments (for example, to `-d` or `--post-data`) should always be given in quotes. If quotes are not used, `&` is interpreted by the shell to indicate that this should be a background process.

How it works...

If you look at the website source (use the **View Source** option from the web browser), you will see an HTML form defined, similar to the following code:

```
<form action="http://book.sarathlakshman.com/lsc/mlogs/submit.php"
method="post" >

<input type="text" name="host" value="HOSTNAME" >
<input type="text" name="user" value="USER" >
<input type="submit" >
</form>
```

Here, `http://book.sarathlakshman.com/lsc/mlogs/submit.php` is the target URL. When the user enters the details and clicks on the **Submit** button, the host and user inputs are sent to `submit.php` as a POST request, and the response page is returned on the browser.

See also

- ▶ The *A primer on cURL* recipe in this chapter explains the `curl` command
- ▶ The *Download a file from a website* recipe in this chapter explains the `wget` command