Exercise 2: Using cURL

In this exercise, you will be using the command-line cURL to make REST calls to the Swagger demo pet store API. We'll make the same calls as for Exercise 1 so that you can easily compare using a GUI tool to using cURL.

Note: Although the official name of the tool is "cURL", most people just call it "curl".

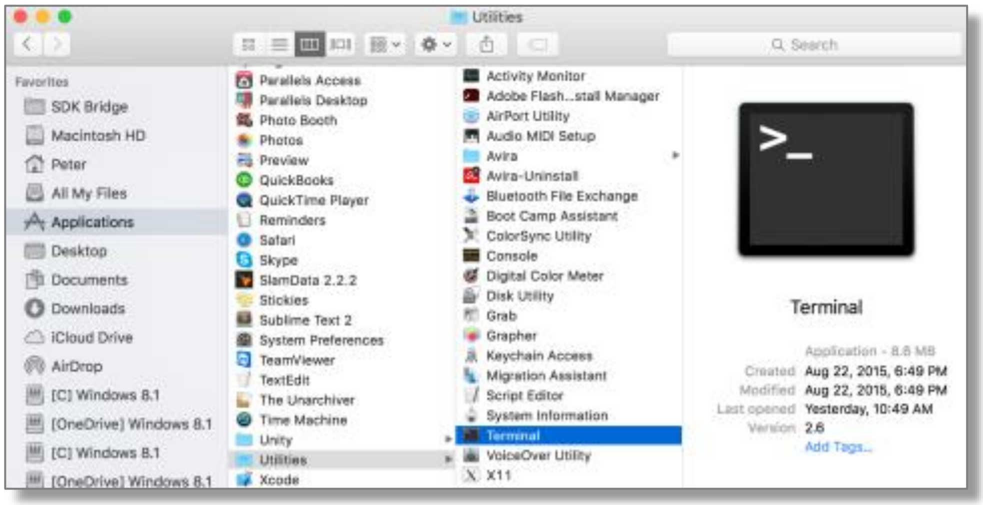
Set up

Follow these steps to install curl.

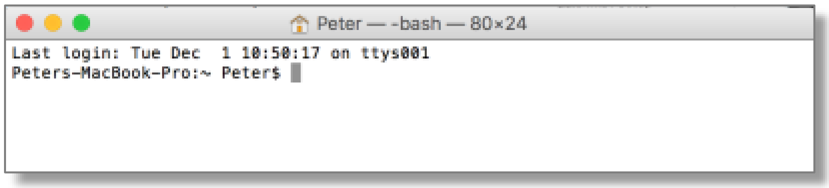
Mac OS X

If you are using Mac OS X, you are fortunate because curl comes installed already. All you need to do is open up the terminal to use it:

1. Open the Finder.
2. Go to **Applications**.
3. Open the **Utilities** folder.
4. Double-click on **Terminal**.



5. The terminal app should open.



```
Last login: Tue Dec  1 10:50:17 on ttys001
Peters-MacBook-Pro:~ Peter$ 
```

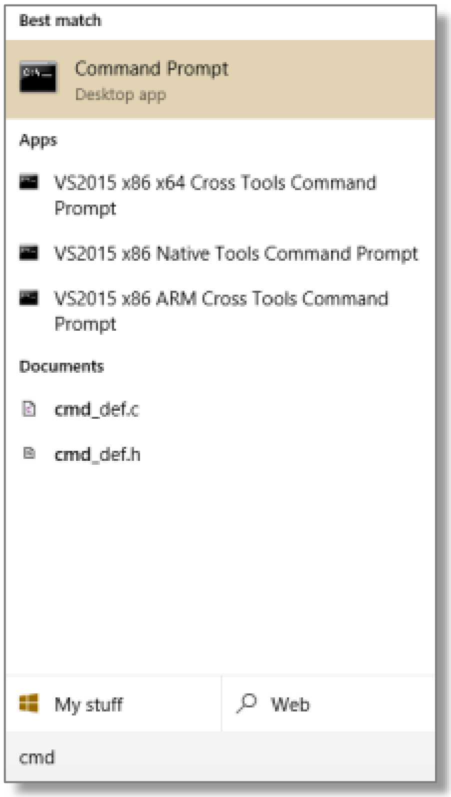
Windows

For Windows, you will need to download the curl executable and put in a place where it can be easily accessed. Follow these steps:

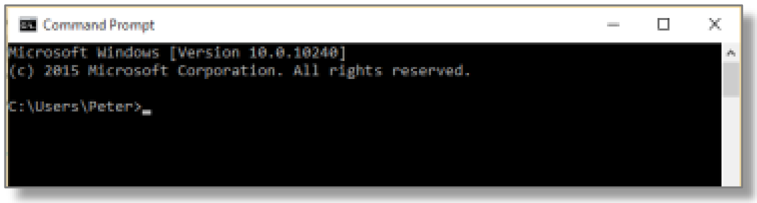
1. In a browser, go to the curl download wizard: http://curl.haxx.se/dlwiz/
2. Click on **curl executable**.
3. Choose your operating system (Win32 or Win64, depending on whether your computer is a 32 or 64 bit machine).
4. Leave the dropdown as **Generic** and click **Select**.
5. Select **Unspecified** for the dropdown and click **Select**.
6. If you are asked, select **x86_64** for the dropdown and click **Select**.
7. Click the top **Download** button for SSL enabled for the version that says **The file is packaged using CAB**. A file called **curl-x.x.x.cab** will appear in your **Download** folder (where the x's are version numbers).
8. Double-click the file to open it.
9. Copy the file **CURL.EXE** to a convenient location, ideally not too many levels deep. I like to put it in my account directory, which is `c:\Users\Peter`. **Note:** For some versions, there will be multiple **CURL.EXE** files. Just pick one. (The second biggest one seems to often work.) When you are using the instructions below for `curl --help`, if you get an error, then try one of the others.

Note: Sometimes the cab version won't open. It depends on the curl and Windows versions. Just try other curl versions on that page and see if you can get the CURL.EXE file another way. Usually there's at least one way that will work.

10. Open the **Command Prompt**, which is the console app. How to do this will depend on your version of Windows. For Windows 10, click on the Start icon and type in `cmd`. Then choose **Command Prompt.**


Best match
Command Prompt
Desktop app
Apps
VS2015 x86 x64 Cross Tools Command Prompt
VS2015 x86 Native Tools Command Prompt
VS2015 x86 ARM Cross Tools Command Prompt
Documents
cmd_def.c
cmd_def.h
My stuff
Web
cmd

11. You should see the command prompt.



```
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Peter>_
```

12. If you are not in the directory where you put curl.exe, then navigate to that directory by typing in:

`cd c:\<The path to your directory>`

For example, to navigate to the **Peter** directory in the **Users** directory, type:

`cd c:\Users\Peter`

13. Type `dir` to get a list of files in that directory. CURL.EXE should be listed.

Trying out curl

The way that curl works is that you type the command `curl` followed by a series of flags that tell curl what to do. Flags can be denoted by either a single dash or a double dash. The single dash uses a onecharacter abbreviation, whereas the double dash uses a longer version. By using the flag help, you can see a list of all possible flags. In the console, type:

```
curl --help
```

You should see a big list of flags and descriptions. Don't worry about understanding them. Just seeing the list means that it's working.

Troubleshooting: If you are on Windows and you see this message:

> ```
> 'curl' is not recognized as an internal or external command, operable program or batch file.
> ```
>
> That means that you are not in the directory where **CURL.EXE** is located.

What you just typed used the double dash and the longer version of the flag that's easy to read. You could do the same with the shorter version of the flag and only one dash:

```
curl -h
```

Try it out.

Making HTTP Requests

Let's make the same HTTP requests that we did for exercise 1 so you can see how to do it in curl.

Create a User

Create a new user in the system. We are going to make a POST request to the URL `http://petstore.swagger.io/v2/user` and include data for the new user in the POST body in JSON format.

The **--request** flag indicates the method. In this case, we are doing a POST.

The **--header** flag indicates a header. The header name and value are in quotes and indicate that the POST body is in JSON and the return body should also be in JSON.

The **--data** flag indicates the POST body, which is followed in quotes. Note that the quotation marks in the JSON need to be "escaped" so that they aren't confused with the end quote for the POST body. "Escaping" means putting a back-slash in front of the quote.

Copy and paste the following line into the console.

```
curl --request POST --header "Content-Type: application/json"
  --data "{ \"id\": 0, \"username\": \"restexpert\",
  \"firstName\": \"Rachel\", \"lastName\": \"Rest\",
  \"email\": \"rachel@example.com\", \"password\": \"json\",
  \"phone\": \"5551212\", \"userStatus\": 0 }"
  "http://petstore.swagger.io/v2/user"
```

This request returns no data, so you won't see anything returned. You'll know it's correct if you do not see any errors.

If you do see errors, it may be that the copy-and-pasting process is adding in line breaks. This will depend on the browser or app that you are using to view this document. Here are some workarounds:

1. For MacOS, use the following command. Note that the backslashes at the end allow you to continue a command line onto multiple lines:

```
curl --request POST --header "Content-Type: application/json" \
  --data "{ \"id\": 0, \"username\": \"restexpert\", \
  \"firstName\": \"Rachel\", \"lastName\": \"Rest\", \
  \"email\": \"rachel@example.com\", \"password\": \"json\", \
  \"phone\": \"5551212\", \"userStatus\": 0 }" \
  "http://petstore.swagger.io/v2/user"
```

2. For Windows, or if the Mac solution didn't work, copy and paste the following command into a text editor and make sure that everything is on one line. Delete any line breaks if you have to. Also, make sure there is a space between the closing quote of 0 }" and the starting quote of "http://.

```
curl --request POST --header "Content-Type: application/json" --data "{ \"id\": 0,
\"username\": \"restexpert\", \"firstName\": \"Rachel\", \"lastName\": \"Rest\", \"email\":
\"rachel@example.com\", \"password\": \"json\", \"phone\": \"5551212\", \"userStatus\": 0 }"
"http://petstore.swagger.io/v2/user"
```

Note: If you wanted to use the shorter abbreviations for the flags, you could have used -X instead of --request, -H instead of --header, and -d instead of --data.

Retrieve User Information

To retrieve user information, we just need to change the POST method to GET, remove the Content-Type header and POST body, and add the username on the end of the URL. Copy and paste the following line into the console:

```
curl --request GET http://petstore.swagger.io/v2/user/restexpert
```

The following will be returned. This is unformatted JSON with all of the user information.

```
{"id":0,"username":"restexpert","firstName":"Rachel","lastName":"Rest","email":"rachel@exa
mple.com","password":"json","phone":"5551212","userStatus":0}
```

Delete the User

To delete the user, simply change the GET method to DELETE:

```
curl --request DELETE http://petstore.swagger.io/v2/user/restexpert
```

If successful, nothing will be returned. Try doing the GET request again and this time you will see JSON returned that indicates that the user is not found.