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".

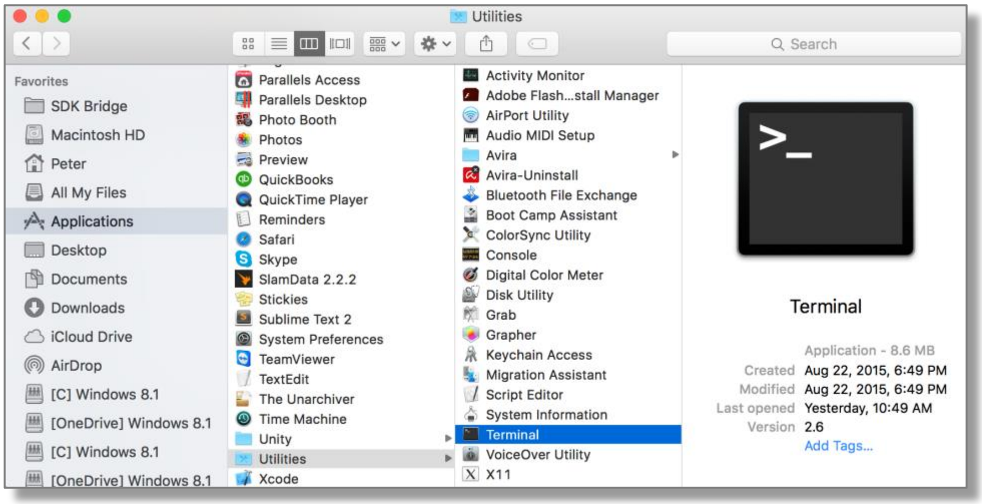
Set up

Follow these steps to install curl.

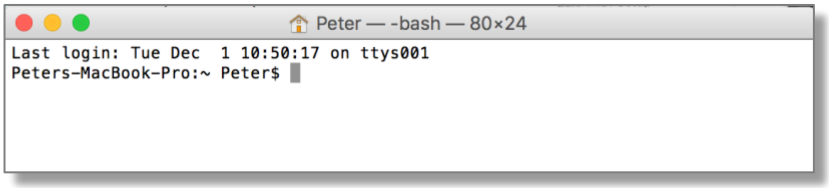
Mac OS X

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.

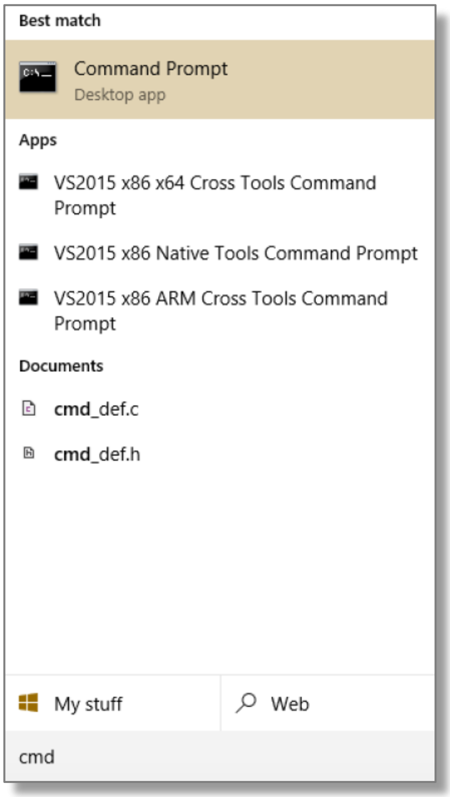


```
Last login: Tue Dec  1 10:50:17 on ttys001
Peters-MacBook-Pro:~ Peter$ 
```

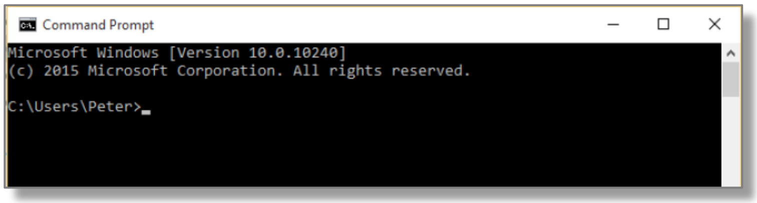
Windows

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 page: https://curl.haxx.se/windows/
2. Click on the 64 or 32 bit version, depending on your operating system.
3. Download and extract the zip file. (Right click and then **Extract All…**
4. In the extracted folder, open the **bin** folder.
5. 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`
6. 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

7. You should see the command prompt.



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

C:\Users\Peter>_
```

8. 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
```

9. 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 one-character 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 Pet

Create a new pet in the system. We are going to make a POST request to the URL **`https://petstore.swagger.io/v2/pet`** and include data for the new pet 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 I used single quotation marks for the data so that I could have double quotation marks inside the JSON. Another option is to “escape” the quotation marks by putting back-slashes in front of them.

Copy and paste the following line into the console. Feel free to change the ID to another 4 digit number and the name to another name. It should all be one line.

```
curl --request POST --header "Content-Type: application/json" --data
"{  \"id\": 5656, \"name\": \"Bela Bardog\", \"status\":
\"available\"}" "https://petstore.swagger.io/v2/pet"
```

You will see your object returned. You’ll know it’s correct if you do not see any errors.

If you see an error that says “SSL certificate problem,” this is because the URL involves the secure protocol https, and your computer doesn’t like Swagger’s certificate. You can tell it to ignore the certificate by adding a flag **–insecure**, like this:

```
curl --request POST --header "Content-Type: application/json" --data
"{  \"id\": 5656, \"name\": \"Bela Bardog\", \"status\":
\"available\"}" "https://petstore.swagger.io/v2/pet"
```

If you do see other 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\": 5656, \"name\": \"Bela Bardog\", \
  \"status\": \"available\"}" \
  --insecure "https://petstore.swagger.io/v2/pet"
```

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 `available\" }"` and the starting quote of `"https://`.

```
curl --request POST --header "Content-Type: application/json" --data
"{  \"id\": 5656, \"name\": \"Bela Bardog\", \"status\":
\"available\"}" -insecure "https://petstore.swagger.io/v2/pet"
```

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 Pet Information

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

```
curl --request GET https://petstore.swagger.io/v2/pet/5656
```

Again, if you get the SSL certificate errors, add the **insecure** flag:

```
curl --request GET --insecure https://petstore.swagger.io/v2/pet/5656
```

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

```
{"id":5656,"name":"Bela
Bardog","photoUrls":[],"tags":[],"status":"available"}
```

Delete the Pet

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

```
curl --request DELETE https://petstore.swagger.io/v2/pet/5656
```

If successful, it will return a code of type 200, which is the HTTP code for success. Try doing the GET request again and this time you will see JSON returned that indicates that the pet is not found.

Once again, add the insecure flag if needed for the DELETE and GET requests.