(COMPUTER) SCIENCE AND COOKING

KEVIN SCHMID

Within the HTTP request-response paradigm, a web browser plays a very important role for the end user. Not only does the browser create HTTP requests, sending cookies and headers appropriately, but it also parses HTTP responses, displaying beautiful, interactive websites that were originally just sent as 1s and 0s.

In a sense, the web browser provides a layer over HTTP for the user: the browser manages the communication so the user doesn't have to. But what if we had to do this ourselves?

Enter `telnet`, a command-line utility that, according to Wikipedia, "provide[s] a bidirectional interactive text-oriented communication facility." For our purposes, we can use `telnet` to send HTTP requests and receive HTTP responses.

So let's try this out with the HarvardFood API since, according to its documentation, "you can query it using simple HTTP GET requests." Cool! (For the remainder of this lab, refer to the API's documentation at
`https://manual.cs50.net/HarvardFood_API`.)

To start telnet and specify that you are connecting to food.cs50.net, enter `telnet food.cs50.net 80` at a terminal, which indicates the server and port number. If everything went well, you should see something like this:

```
francis@appliance:~ telnet food.cs50.net 80
Trying 140.247.63.236...
Connected to hs.cs50.net.
Escape character is '^]'.
```

And now you are free to enter your request! Let's start by entering a simple request, something like we saw in lecture:

```
GET / HTTP/1.1
Host: food.cs50.net
```

Type this at the prompt (and press enter twice.) You should see a response from the server like this:

```
HTTP/1.1 302 Moved Temporarily
Date: Thu, 30 Aug 2012 14:47:15 GMT
Server: Apache
Location: https://manual.cs50.net/HarvardFood_API
Content-Length: 0
Connection: close
Content-Type: text/html; charset=UTF-8
```

And then telnet should report:

```
Connection closed by foreign host.
```

Some questions:

(1) Visit `http://food.cs50.net` in a browser. What kind of behavior is exhibited when you visit this website, and is it consistent with the response you get in telnet? What header(s) indicate this?

(2) (a) Using the API documentation, figure out what URLs will provide nutrition facts for 1 portion of recipe 117003.
    (b) Request the data in CSV and JSON format using telnet. What are the differences in the headers sent in the response, and how does this affect the behavior that you see when visiting these URLs in the browser?
(3) Why do you think that it's necessary to enter the `Host :` in the request?
(4) A possible extension: Send an email message using telnet.