

Exercise 3

 coursys.sfu.ca/2021fa-cmpt-470-d1/pages/Exercise3

For this exercise, you will explore the technology in the course and the basics of HTTP conversations. Create a file `exercise3.txt` containing your answers to the questions below.

Telnet HTTP Conversations

We will start to explore HTTP connections with the most basic tool available: a Telnet connection to port 80. Telnet is essentially a TCP client: it makes a TCP connection and you can send and receive characters.

You should be able to do this with any Telnet client; on Linux/Unix, the connection can be made like this:

```
telnet hostname 80
```

First request the URL <http://www.sfu.ca/~ggbaker/470e3/> by telnetting to `www.sfu.ca` on port 80 and making a request like this (with a trailing blank line to indicate the end of the request headers):

```
GET /~ggbaker/470e3/ HTTP/1.1
Host: www.sfu.ca
Connection: close
```

[Depending on the server's timeout, you may need to form your requests in a text editor and then copy-and-paste.]

In your `exercise3.txt` file, indicate what the status code was for the server's response.

If-modified-since

Repeat the request adding an `If-modified-since` header to the request, which will simulate an existing cached version of the page:

```
If-modified-since: Wed, 1 Sep 2021 00:00:00 GMT
```

In your `exercise3.txt` file, indicate what the status code was for the server's response and how the response differed from the previous question.

Redirect

To see a redirect in action, request the URL <http://www.sfu.ca/~ggbaker/470e3> (**without** the trailing slash) as above.

In your `exercise3.txt` file, indicate what the status code was for the server's response and how the response differed from the original request.

Command-Line Requests

Try **at least one** of these command-line tools (`curl` or `GET`). In your `exercise3.txt` , indicate the difference the command-line options (`-I` or `-esS`) made in the output of the command.

Use `curl` to request these URLs:

```
curl http://www.sfu.ca/~ggbaker/470e3/
curl -I http://www.sfu.ca/~ggbaker/470e3/
curl -I http://www.sfu.ca/~ggbaker/470e3
```

Have a look at the output to see the information displayed (especially for the more verbose version with and without the redirect).

Use `GET` to request these URLs:

```
GET http://www.sfu.ca/~ggbaker/470e3/
GET -esS http://www.sfu.ca/~ggbaker/470e3/
GET -esS http://www.sfu.ca/~ggbaker/470e3
```

Have a look at the output to see the information displayed (especially for the more verbose version with and without the redirect).

Developer Tools

Use either the [Firefox developer tools](#), or the [Chrome developer tools](#) to view the request for a few web pages. (Open the Network tab before the request, and expand to see the request and response headers.)

Also have a look at the display of the HTML source and associated CSS rules. (In the Inspector/Elements tab, expand some elements to explore the markup.)

In your `exercise3.txt` , write “I promise I played with the developer tools”.

Git Intro

We will be using Git as a version control system for this course. This is a good time to get the your accounts and tools set up. I have provided some [instructions on our GitLab setup](#).

First, get yourself set up with GitLab: [log in](#) (with “SFU ID”) and get your SSH key (created if necessary and) associated with your GitLab account.

Create a repository for your weekly exercise work: create a project with visibility level “Private” and the name “exercise-3”.

Add the instructor and TA as members of the project: if you don't, we won't be able to see your work to mark it. In the GitLab project, click “Settings”, “Members”, and “New project member”, and add both of us (ggbaker, lirongl, gna17) with “Developer” access.

If you don't add us as developers on the repository, we have no way to give you marks. This will be true of all future Git repositories as well: no access = no marks.

Working with Git

Now start working with your Git client. (I'm going to assume Linux-style command line Git interaction. If you're working with a graphical Git client, translate appropriately.) First make sure the client knows who you are (substituting your name and SFU email):

```
git config --global user.name "Your Name"
git config --global user.email "USERID@sfu.ca"
```

Bootstrap your repository by following the “Create Repository” instructions on your project page in GitLab (`mkdir exercise-3` ; `cd exercise-3` ; `git init` ; ...). After this, if you click on the “Files” tab for your project in GitLab, you should see the `README` file that you created and pushed. [Note that the `git init` process is once-per-repository. If you want to work on the same repo on another computer, you just need to `git clone` it.]

Note: If you're asked for a password for the “git@csil-git1.cs.surrey.sfu.ca” account, then your SSH key isn't being used by your client. There is no password for that account: fix your SSH key.

Edit the `README` file and add a sentence about what this repository is for.

Copy your `exercise3.txt` into the cloned repository (i.e. the `exercise-3` directory). You can add it to the repository and push your changes to the server like this:

```
git add exercise3.txt
git commit -a # please write a good commit message, here and always
git push
```

You should now see the updated `README` and the new `exercise3.txt` in the “Files” tab in the GitLab project. If not, something has gone wrong and your files won't be there to mark.

You might also want to spend a few minutes learning more about Git:

Git Tag

Submission of Git contents will be by giving us a `git tag`, which is basically an immutable snapshot of the repository. That gives us a clear target for marking.

You can create a tag “e3submission” for this exercise like this:

```
git tag e3submission  
git push --tags
```

We will check out that tag for marking. You will submit the repository checkout URL (like `git@csil-git1.cs.surrey.sfu.ca:USERID/exercise-3.git`) and the name of the tag.

Submitting

Submit your Git tag through CourSys for Exercise 3.

Updated Mon Aug. 30 2021, 07:36 by ggbaker.