

# CS2910 Lab 3: HTTP Client

---

## Lab Assignment

This is a team assignment; each team should be two members unless a different size is approved by the instructor.

## Introduction

The goal of this lab is to write a short Python program, to request and save a web resource, acting as an HTTP client.

The program has at least the following functions; you may add others.

### **main()**

The provided main function will perform basic tests. You may add others. No user input is needed.

- This method has no arguments
- This method is invoked with a **main()** function call at the end of the program.

### **get\_http\_resource(url,file\_name)**

Using HTTP, request a web resource and store the returned data in the specified file.

- Arguments:
  - **url**: string containing URL for desired resource
  - **file\_name**: string containing name of file in which to store response data
- Return value:
  - String "OK, requested entity received, nnn bytes in size" or error message
- Operation:
  - Send an HTTP request to retrieve the resource at the specified **url**.
  - If successful, store the response data in a file with the specified **file\_name**.
  - Return the specified status information as a string.

## Procedure

1. Download the skeleton Python template: [httpclient.py](#)
2. Edit the header of the file to include your team members' names.
3. Complete the method to request, receive, and store the designated resource. You may add other helper methods, but do not change the code provided in the template.
4. Add comments *at the end* of your Python file, with the following information:
  - A description of the functionality you implemented and the results of your testing.
  - Comments on your experience in completing the lab, including any problems you encountered. Briefly explain what you learned.
  - Questions and suggestions.

Additional implementation details will be discussed in class. If you have questions about these requirements, ask in class or lab.

If this base functionality turns out to be too easy, you may experiment with adding additional functions, but be sure the basic requirements are still met.

Try to divide up the primary responsibility for parts of the program in an equitable way.

## Submission (Due Thursday, Week 4, 11PM)

One team member should submit your Python file by uploading it to the upload page (which will be linked from the Schedule).

## Rework Submission (Due electronically Friday, Week 5, 11PM)

You may submit a reworked submission to regain partial credit for your Lab 3 submission. If you are OK with your Lab 3 work and grade, you need not submit anything from this "rework" exercise. Just apply any lessons to the next lab.

If you rework the lab, it must include a design in the format of the in-class exercise we worked yesterday (Monday of Week 5). Do this design *first*. It is more important to the rework than anything else. As you work the design, think about how you can interpret the HTTP response before reading the whole document into memory.

The format of the design strategy we discussed in class on Monday was as follows:

- Write the headers for the methods – that is, the first line of the method, e.g.

```
def method_name(argument1, argument2):
```

- Write documentation for each header, i.e. the comment above the header. This should give the “user’s perspective of what the method does, for example:

```
# This method fetches a resource from the server and saves it to a file.
#
# Does not save file if an error occurs - instead, prints an error to the screen
def method_name(argument1, argument2):
```

- Write comments (*not* code) in the body of each method describing the major steps that method does to meet its goals.

```
# ..
def method_name(argument1, argument2):
# Connect to server
# Send Request
# ...
```

Submit the lab electronically through the website. If you turned your submission in late, please wait a couple of days for me to download it before uploading the reworked submission.

(Acknowledgements: The original version of this lab written by Dr. Sebern.)