

CS361 COMPUTER NETWORKS LABORATORY

Computer Networks Project

Project #11: Browser

Milind Pandey (201851068)

Prateek Senapati (201851090)

Sudhanshu Pandey (201851131)

Sumit Kumar Mishra (201851133)

Problem Statement

Create a basic HTTP browser which is capable of sending HTTP HEAD, GET, POST, PUT, DELETE requests as per the specification given in RFC-2616. The browser must pass test cases after getting connected to a standard HTTP server (e.g. Apache web server). Do not write your own server to test the application. An implemented server from another group's project can also be used to test the client.

Project Analysis

Our project idea was to create a basic HTTP requests checker, which would allow a user to send HTTP requests (HEAD, GET, POST, PUT, DELETE) to any standard HTTP server and get the appropriate responses in return. The user will be able to send URL parameters (in the case of GET and HEAD requests) and form data and files (in the case of POST and PUT requests) along with the HTTP requests. Our HTTP requests checker application will fetch responses from servers and display the URL, the status code and the server's response headers.

Design

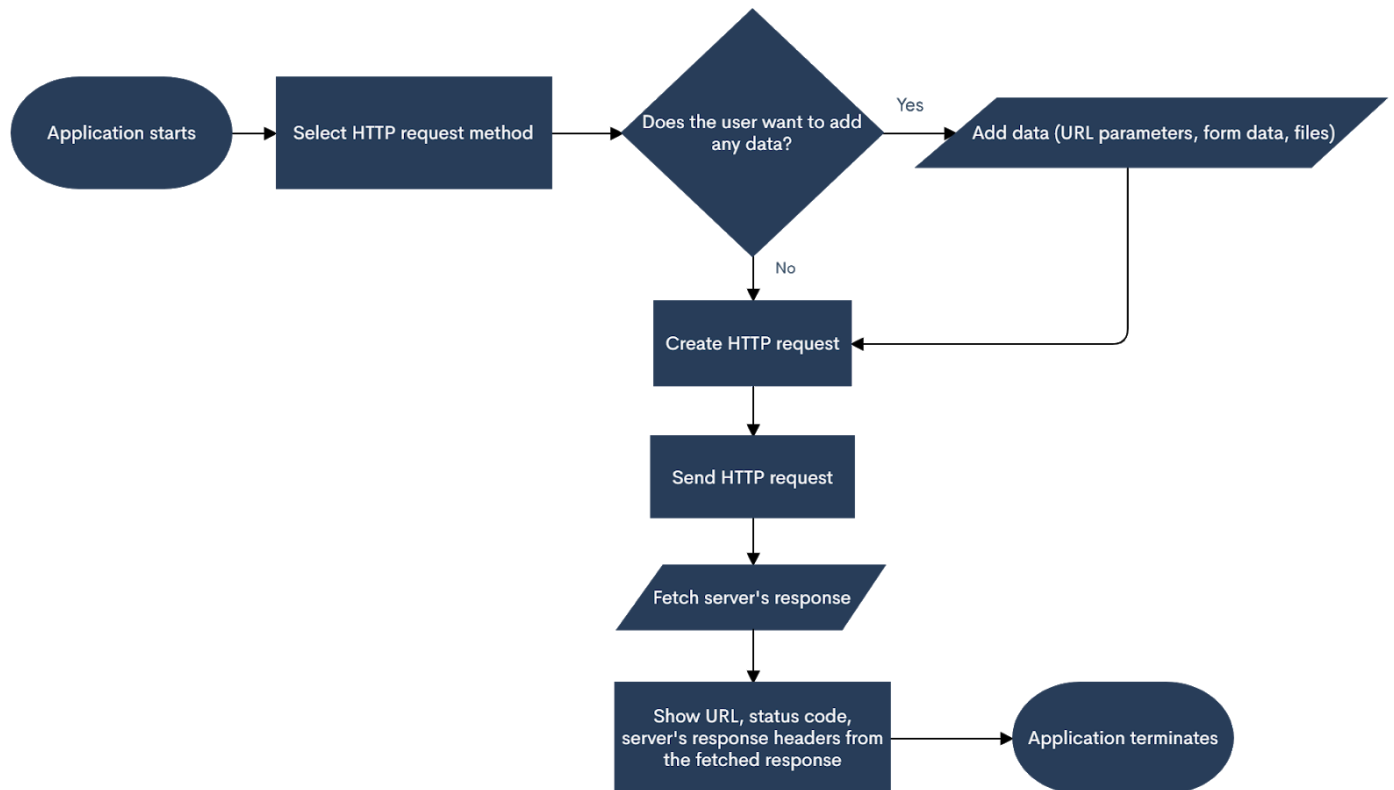


Figure 1: Flowchart of our browser (HTTP requests checker) application

Requirements

- Python version 3.6 or 3.7
 - There are some issues when downloading and installing *streamlit* with Python 3.8 and Python 3.9. We have tested the complete process of working with our application using Python 3.6 and Python 3.7 and it works fine. Therefore, we highly recommend using Python 3.6 or Python 3.7 for working with our application.
 - You can download Python from [here](#).
 - You can refer to this [link](#) for Python download and installation instructions.
- [requests](#) (*requests* is an HTTP library for Python)
 - After installing Python, you can install *requests* by going to your terminal or command prompt and running the command: `pip install requests`.
- [streamlit](#) (*streamlit* is a framework used to create Python web apps)

- After installing Python, you can install *streamlit* by going to your terminal or command prompt and running the command: `pip install streamlit`.

Project Relevant Information

- httpbin.org is a good resource to test the working of an HTTP requests checker (or, a simple browser). Corey Schafer has a very concise YouTube tutorial on using the *requests* library with *httpbin.org*. You can watch the video [here](#).
- We have also included a short screencast/video of the working of our browser (HTTP requests checker) application in the **Outputs** section below. This screencast can be referred to get more information on how to use our application.
- In the active application interface (more information on this in the following section), you have to manually clear the input area/space for URL parameters, form data and file data before adding details for a new input (and after adding details of the previous input). Please watch the screencast of our application in the **Outputs** section below to get a better idea on how to do this.

Steps to Run the Application

- Download the zipped project folder and unzip it. The project folder contains two files: this final report and a Python script named **browser.py**.
- Open your terminal or command prompt.
- In your terminal or command prompt, go to this downloaded and unzipped project folder.
- One way to check if you are in the correct directory location is to run the command `dir` in the command prompt (`ls` in the terminal). If everything is going right so far, you will be able to see **browser.py** in the result of the `dir` or `ls` command.
- Once you are in the correct directory location, run the following command: `streamlit run browser.py`. This will open the browser (HTTP requests checker) application in your default browser. You can also use the Local URL or the Network URL given in the output of this command to open the application in a browser of your choice.

- To stop the application, enter **CTRL+C** in your command prompt or terminal. This will stop the application but it will still be visible in your browser in a deactivated state. You can close the tab to close the deactivated application interface. Also, closing the tab in which the application interface is active does not stop the application from running in an active state. To properly stop the application, it has to be deactivated using the **CTRL+C** command.
- In the active application interface, you can select the HTTP request method. Based on your selection, you will be prompted to add URL parameters, form data or files. Once you have added the required parameters or data, you can send the HTTP request using a button (buttons are added at appropriate places to facilitate the working of the application). Important details (URL, status code and server's response headers) from the fetched response will be displayed on the screen. For more details on how you should use our browser (HTTP requests checker), you can watch the screencast of our application in the **Outputs** section below.

Outputs

You can find the screencast of the working of our browser application here:

<http://bit.ly/httprequestschecker>. In this screencast, you can see the outputs of some of the test cases we have tried on our application. Please ensure that you are using your IIIT Vadodara Google account to watch this video (view access has been given to only the members of the IIIT Vadodara group on Google).

Note: As no test cases were given in the problem statement, we tested our application by creating our own test cases with the help of *httpbin.org*.