Open-Source Technology Use Report: HTTP Parsing

[Flask]

General Information & Licensing

Code Repository	https://github.com/pallets/flask
License Type	BSD-3-Clause
License Description	• The BSD-3-Clause allows for the redistribution or modification of the flask source code as long as a copyright notice, condition list, and disclaimer accompany the distribution.
License Restrictions	• With this license, users cannot create products with this software then promote/endorse those products using the copyright holder's name or the names of any of its contributors without prior written permission.
Who worked with this?	To be decided

Purpose

Flask allows us to parse the HTTP headers of incoming requests using the @app.route decorator. @app.route(rule="path",methods=["method"]) allows us to execute a particular function whenever we get a request of the "method" type at the specified "path". This removes the need for us to parse the raw HTTP request data coming from the TCP socket as we did in the homeworks. This method is used several times in the app.py file in order to route each of the requests we expect to get. For example, if the server receives a GET request to the homepage at the path "/", @app.route("/", *methods* = ["GET"])handles this request.

Magic ⋆★˚｡⋆˚☽˚｡⋆˚★⋆

The run method from Flask class does the following to establish a server:

1. It imports the run_simple method form the werkzeug library and calls it here
2. The run_simple() method calls make_server() here.

When the server is set up using the make_server() function, request_handler is none. make_server() returns an instance of BaseWSGI server here which assigns handler to an instance of the WSGIRequestHandler class since handler is none here. This handler variable will then tell the server how to handle incoming TCP requests since we call the constructor of the TCPServer class here using handler (this is because WSGIRequestHandler inherits from HTPPServer which inherits from sockerserver.TCPSever here)

The WSGIRequestHandler class inherits from http.server.BaseHTTPRequestHandler here. Since BaseHTTPRequestHandler inherits socketserver.StreamRequestHandler here which inherits from socketserver.StreamRequestHandler here, whenever there is an incoming HTTP request on a TCP socket socketserver.BaseRequestHandler calls self.handle() here. Once this call is made, the following takes place so that the HTTP header is parsed:

1. The socketserver.BaseRequestHandler.handle() method calls the WSGIRequestHandler handle() method which calls the http.server.BaseHTTPRequestHander.handle() method here.
2. http.server.BaseHTTPRequestHander.handle() calls self.handle_one_request() here. handle_one_request() then calls self.parse_request() here. This is where the path and request type are parsed and stored into self.path and self.command respectively here.
3. parse_request() also parses the rest of the headers using the http.client.parse_headers() method here.