Open-Source Technology Use Report: TCP Connections

[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 copy right notice, conditionlist, 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 build a server that establishes TCP connections with clients. To do this, we create a variable called app, which sets up the flask server (i.e: `app = Flask(__name__)`. This is done in the app.py file in the root directory of our project. This stores an instance of the Flask class.

We then call the run method of the flask class in the main function in the same app.py file (i.e: `app.run(debug=False, host='0.0.0.0', port=8000)`. This sets up the server and runs the application. This main method establishes a server that listens for connections on local port 8000. The server then allows us to form TCP connections with in-coming requests. Once we receive these requests, the flask server parses these requests and calls the app.route functions in our code to respond to the requested path. Flask provides TCP Connections that make it quick to handle requests.

Magic ★★。゜・゜☽゜⌒✧。゜★彡✧༄

The run method from the 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() which returns an instance of the BaseWSGI server class here and stores it in a variable called srv.
3. The BaseWSGI server class constructor enables the server to begin listening for incoming TCP connections by calling self.server_bind() and self.server_activate() if fd is none here.
4. Since BaseWSGI inherits from HTTPServer, BaseWSGI.server_bind() calls HTTPServer.server_bind() which calls socketserver.server_bind() here which binds the server to an address.
5. self.server_activate() from BaseWSGI calls the socketserver.server_activate() method since BaseWSGI inherits from HTTPServer which inherits from socketServer.TCP server here. This function allows the server to begin listening for incoming TCP connections.
6. srv.serve_forever() is then called here which calls the super().serve_forever() method here. Because the BaseWSGI server class inherits from the HTTPServer class (here) and the HTTPServer class inherits the socketServer.TCPServer class here, this is effectively a call to the serve_forever() method from the socketServer.TCPServer class which is what we used on homeworks to create TCP connections with incoming requests.