# Open-Source Report for TCP Connections

## [Flask]

### General Information & Licensing

Code Repository	https://github.com/pallets/flask.git
License Type	BSD
License Description	Framework is available to use for free with or without modifications as long as none of the restrictions below are violated
License Restrictions	<ul> <li>Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.</li> <li>Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.</li> <li>Neither the name of python-flask nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.</li> </ul>



- This line will call a function within file app.py called \_\_init\_\_ which is responsible for binding all the ports and paths to this point. This code can be referenced here:
   <a href="https://github.com/pallets/flask/blob/066a35dd322f689ec07d7c0e82b19eacadac3c6b/src/flask/app.py#L553-#L720">https://github.com/pallets/flask/blob/066a35dd322f689ec07d7c0e82b19eacadac3c6b/src/flask/app.py#L553-#L720</a>
- Within the previous call after the completion of the init call and the server configurations are set up (port, host, directories, rules, environment, etc...) once the server is properly configured for start up the function within app.py \_\_call\_\_ is invoked which can be referenced here:
   <a href="https://github.com/pallets/flask/blob/066a35dd322f689ec07d7c0e82b19eacadac3c6b/src/flask/app.py#L2546">https://github.com/pallets/flask/blob/066a35dd322f689ec07d7c0e82b19eacadac3c6b/src/flask/app.py#L2546</a>
- Now the call function is simply calling the wsgi\_app function and passing parameters generated in the previous call.
- Once inside of werkzeug.serving.py a base wsgi server object is created that abstracts socketServer's handle method to take in data input streams while checking for errors and handling them as necessary. This can be referenced here: <a href="https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/serving.py#L651-#L659">https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/serving.py#L651-#L659</a>
- Once our app has been initialized, it will contain a reference to a BaseWSGIServer object that is built on a socketserver to handle input and output data streams. The WSGI server will be equipped with a request handler class called WSGIRequestHandler that will implement wsgi request/response dispatching for the app. Inside the request handler is a method 'write()' that handles writing the response data to the sockets data stream and is called by the execute() method in WSGIRequestHandler.
- In the same class a call to a class in the same file called DechunkedInput is made, this class contains methods read\_into() and read\_chunk\_length that are used to read the raw data from the input stream.
- After the input stream has been read the data is passed to be interpreted as HTTP
- When responding, the WSGI server waits to be invoked with a properly formatted HTTP response, this response with be sent across the data stream to the client using the methods write() and execute() mentioned above.

#### DechunkedInput:

https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/serving.pv#L86

- read\_chunk\_length() declaration:
   <a href="https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff2">https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff2</a>
   <a href="https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff2">https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff2</a>
   <a href="https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff2">https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff2</a>
   <a href="https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff2">https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff2</a>
   <a href="https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff2">https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff2</a>
   <a href="https://github.com/pallets/werkzeug/serving.py#L97">https://github.com/pallets/werkzeug/serving.py#L97</a>
- readinto() declaration:

https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff2 1f1a/src/werkzeug/serving.pv#L107

#### WSGIRequestHandle:

https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/serving.py#L148

- write() declaration:
   <a href="https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff2">https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff2</a>
   1f1a/src/werkzeug/serving.pv#L248
- execute() declaration:
  - https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff2 1f1a/src/werkzeug/serving.py#L319
- Call to write()

#### Stack trace from app start to page load on client side:

\_\_call\_\_, app.py:2548

https://github.com/pallets/flask/blob/066a35dd322f689ec07d7c0e82b19eacadac3c6b/src/flask/app.pv#L2546

The previously created wsgi server is retrieving the application we have created and wrapping it for its application.

execute, serving.py:322

https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/serving.py#L319

This function is checking to see if the headers have been created yet or not. If they have been created it will send the response. If they have not yet been created it will create the headers accordingly and then proceed to send the proper response.

run\_wsgi, serving.py:335

https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/serving.py#L237

Check if the headers that it has been sent are correct if they write a 100 level response and continue. If they are not correct, set the status and header values to None.

handle\_one\_request, server.py:415

https://github.com/python/cpython/blob/91a8e002c21a5388c5152c5a4871b9a2d85f0fc1/Lib/http/server.py#L391

This method is used to read and respond to a single request on the socket. This will parse the request and retrieve useful information like content length, body, etc.

handle, server.py:427

https://github.com/python/cpython/blob/91a8e002c21a5388c5152c5a4871b9a2d85f0fc1/Lib/http/server.py#L428

Purpose of this method is to handle multiple requests at once. It achieves this by calling the handle one request method multiple times.

handle, serving.py:363

https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/serving.pv#L358

Checks to see if there is a dropped connection. In the case of a dropped connection this method will ignore the dropped connection.

\_init\_\_, socketserver.py:747

https://github.com/python/cpython/blob/91a8e002c21a5388c5152c5a4871b9a2d85f0fc1/Lib/socketserver.py#L748

Sets up BaseRequestHandler class variables, and attempts to call socketservers handle method or finish the servers duties.

finish\_request, socketserver.py:360

https://github.com/python/cpython/blob/91a8e002c21a5388c5152c5a4871b9a2d85f0fc1/Lib/socketserver.py#L358

Calls self.RequestHandlerClass to take care of finishing up post processing of the request.

process request thread, socketserver.py:683

 $\underline{https://github.com/python/cpython/blob/91a8e002c21a5388c5152c5a4871b9a2d85f0fc1/Lib/socketserver.py\#L696}$ 

Attempts to finish a given request or accept and handle any raised errors.

run, threading.py:910

https://github.com/enthought/Python-2.7.3/blob/69fe0ffd2d85b4002cacae1f28ef2eb0f25e16ae/Lib/threading.py#L501

Invokes a callable object to represent the threads activity, attempts to send to target with given keys/args

\_bootstrap\_inner, threading.py:973

https://github.com/enthought/Python-2.7.3/blob/69fe0ffd2d85b4002cacae1f28ef2eb0f25e1 6ae/Lib/threading.pv#L533

Sets up threading class variables and attempts to start thread for communicating with the socket.

\_bootstrap, threading.py:930

https://github.com/enthought/Python-2.7.3/blob/69fe0ffd2d85b4002cacae1f28ef2eb0f25e16ae/Lib/threading.py#L510

Wrapper method for bootstrap that prevents unseen exceptions caused by threads waking up after their apps context has been removed.

