server.HTTP

What It accomplishes:

This Library defines classes to implement HTTP server. In particular, we are using the SimpleHTTPRequestHandler to parse HTTP requests and submit responses back to the client.

How it accomplishes it:

- send_header(self, keyword, value)
    - This function adds a header to the http response being sent
    - The keyword represents the key in the key-value pairs (ex. “Content-Type”)
    - The value represents the value in the key-value pair (ex. “text/html”)
    - The function takes the keyword and value and creates a proper string separated by a “:” and ending with a “\r\n” character, and then appending it to a buffer titled headers_buffer, which stored the headers to the current response being created
    - Github Permalink https://github.com/python/cpython/blob/fa03efda3dc6ad118788bebc61079cd481c0b24c/Lib/http/server.py#L511

- send_response(self, code, message (optional))
    - This function is used to create the header for the HTTP response
    - It first calls the log_request function, which prints a statement declaring the client address, the current date and time, and the type of response sent
    - It then calls send_response_only() which appends the header containing the protocol version (ex: HTTP/1.1), the response code itself, and the message (ex. “OK”, “Redirect”, etc.)
    - It then calls send_header() twice to add headers containing the version string for the server and the current date and time
    - Github Permalink https://github.com/python/cpython/blob/fa03efda3dc6ad118788bebc61079cd481c0b24c/Lib/http/server.py#L484

- end_headers()
    - This function is called to indicate the end of the response header
    - It adds the final “\r\n” to the headers_buffer to indicate the end of the http header
    - It then calls flush_headers(), which takes the headers_buffer and writes it into the wfile, which is the binary output file for the server, and then empties the buffer.
    - Github Permalink https://github.com/python/cpython/blob/fa03efda3dc6ad118788bebc61079cd481c0b24c/Lib/http/server.py#L525

- headers.get(self, name, failobj=None)
    - Used to get a header value from an http request

    - Headers is an object variable in the baseHTTPRequestHandler that is of type Message in the module email
    - The Message class has a value "_headers", which baseHTTPRequestHandler uses to store the headers to the http request.
    - When get() is called, it checks if the value is name is a header that exists in the keys of self._headers
        - If it is, it runs header_fetch_parse() to call sanitize_header(), which checks to return the value as a header value
        - header_fetch_parse(): https://github.com/python/cpython/blob/fa03efda3dc6ad118788bebc61079cd481c0b24c/Lib/email/_policybase.py#L311
        - sanitize_header(): https://github.com/python/cpython/blob/fa03efda3dc6ad118788bebc61079cd481c0b24c/Lib/email/_policybase.py#L281
        - If it does not, it returns the fail object, which is defaulted to None.
    - Github permalink https://github.com/python/cpython/blob/fa03efda3dc6ad118788bebc61079cd481c0b24c/Lib/email/message.py#L462

- self.path
    - Global variable for baseHTTPRequestHandler which stores the path for the request being parsed
    - This is set when the parse_request function is called, which is called in the handle_one_request() function, which is called in the handle() function
        - handle_one_request(): https://github.com/python/cpython/blob/fa03efda3dc6ad118788bebc61079cd481c0b24c/Lib/http/server.py#L386
        - parse_request(): https://github.com/python/cpython/blob/fa03efda3dc6ad118788bebc61079cd481c0b24c/Lib/http/server.py#L269

- header.get_boundary()
    - Similar functionality to header.get()
    - Calls get_param(), setting param, to "boundary", which calls _get_params_preserve(), which parses through the "Content-Type" header by returning a list split by the ";" in content-type, with each element in the list being a tuple split by "="
    - If boundary is found, then we return the value for boundary. Otherwise we return a fail object, which is defaulted to None
    - Permalinks:
        - get_boundary(): https://github.com/python/cpython/blob/fa03efda3dc6ad118788bebc61079cd481c0b24c/Lib/email/message.py#L822

- get_param(): https://github.com/python/cpython/blob/fa03efda3dc6ad118788bebc61079cd481c0b24c/Lib/email/message.py#L667
- _get_params_preserve(): https://github.com/python/cpython/blob/fa03efda3dc6ad118788bebc61079cd481c0b24c/Lib/email/message.py#L624

Licences/ Terms of service:

This module was used under the python standard library, so the license agreement is under the Python Software Foundation (PSF), which allows the licensee nonexclusive and royalty-free access to its libraries to display publicly. It is also licensed under the zero-clause BSD license, which allows the user to copy, modify, or distribute the software with or without fee.