CSE312 Parsing HTTP headers Report

Web Framework: Flask / Python

General Information & Licensing

Code Repository	https://github.com/pallets/werkzeug/tree/3115aa6a6276939f5fd6efa46282e0256ff21f1a
License Type	rst
License Description	Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. 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. 3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
License Restrictions	THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT

	OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Magic ⋆★⁺₊˚⋆˚☽˚⌒➷₊˚★彡⁺★༄

Dispel the magic of this technology. Replace this text with some that answers the following questions for the above tech:

- How does this technology do what it does? Please explain this in detail, starting from after the TCP socket is created
- Where is the specific code that does what you use the tech for? You ***must*** provide a link to the specific file in the repository for your tech with a line number or number range.
  - If there is more than one step in the chain of calls *(hint: there will be)*, you must provide links for the entire chain of calls from your code, to the library code that actually accomplishes the task for you.
  - Example: If you use an object of type HttpRequest in your code which contains the headers of the request, you must show exactly how that object parsed the original headers from the TCP socket. This will often involve tracing through multiple libraries and you must show the entire trace through all these libraries with links to all the involved code.

In this project, our group using built-in function that help parsing HTTP headers. Specifically, we are using **request** function from **flask** library which help get the information from the HTTP headers.
We are using request.form, request.method, request.path, request.cookie, request.files

This is how request.cookie work when we call them.
Request.cookie ->

- CSE312_GroupProject-/Server/project_server.py (Line 123)
- Link: https://github.com/XuyangLi-12088/CSE312_GroupProject-/blob/master/Server/project_server.py#L123

-
  ```
  print("Request header: " + str(request.headers), flush = True)
  ```
- Set headers to a Headers type (line 127)
    - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L127
- Set self.headers = headers (line 148)
    - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L148
- Call access_route(self) function (line 188)
    - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L188
- Access_route function checks if a forwarded header exists a list of all ip addresses from the client ip to the last proxy server.
    - If “X-Forwarded-For” in self.headers
    - Call parse_list_header(self.headers["X-Forwarded-For"]) (line 307)
        - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L307
        - parse_list_header() function will parse a list with all comma-separated elements. Those elements may include quoted-string. A non-quoted string could have quotes in the middle.
        - Call parse_http_list() function (line 1431)
            - Link: https://github.com/python/cpython/blob/5ea052bb0c8fa76867751046c89f69db5661ed4f/Lib/urllib/request.py#L1431
            - parse_http_list() function will do the actual parsing work, parse comma-separated lists where the elements of the list may include quoted-strings.
        - Call unquote_header_value() function (line216)
            - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py

#L216

- unquote_header_value() function will remove quotes from a header value
- Call host(self) function (line 238)
    - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L238
    - host() function will get the host name from the request that was made.
    - Call get_host() function (line 56)
        - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/sansio/utils.py#L56
        - get_host() function will get the “host” header value, with port if necessary.
- Call cookies(self) function (line 247)
    - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L247
    - cookies() function will get all cookies from the header
    - Call parse_cookie() function (line 97)
        - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/sansio/http.py#L97
        - parse_cookie() function will parse a cookie from string, and return a dictionary
        - Call _cookie_parse_impl() function (line 381)
            - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/_internal.py#L381
            - _cookie_parse_impl() function is a low-level cookie parsing facility that operates on bytes.
- Call content_length(self) function (line 270)
    - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L270
    - content_length() function will get the content length from the request header, content length indicates the size of the entity-body in bytes.
- Call _parse_content_type(self) function (line 344)
    - Link:

https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L344

- _parse_content_type() function will get the content type of the header
  - From self._parsed_content_type, in mimetype(self) function and mimetype_params(self) function (line 351 and 361)
    - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L351
    - https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L361
    - mimetype() function and mimetype_params() function will get the mime type and mime type parameters from the header.

- Call pragma(self) function (line 370)
  - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L370
  - pragma() function will get the Pragma general-header field from the header.
  - Call parse_set_header() function (line 608)
  - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L608
  - parse_set_header() will parse a set-like header
    - Call parse_list_header() function (line 307)
      - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L307
      - parse_list_header() function will parse a list with all comma-separated elements. Those elements may include quoted-string. A non-quoted string could have quotes in the middle.
      - Call parse_http_list() function (line 1431)
        - Link: https://github.com/python/cpython/blob/5ea052bb0c8fa76867751046c89f69db5661ed4f/Lib/urllib/request.py#L1431
        - parse_http_list() function will do the actual parsing work, parse comma-separated lists

where the elements of the list may include quoted-strings.

- Call accept_mimetypes(self) function (line 382)
    - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L382
    - accept_mimetypes(self) function will get the list of accept mime types from this client from header
    - Call parse_accept_header() function (line 471)
        - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L471
        - parse_accept_header() function will parses an HTTP Accept-* header.
        - “_accept_re” is doing the actual parsing using the Compile() function, which will compile a regular expression pattern, returning a Pattern object.
            - Link: https://github.com/python/cpython/blob/5ea052bb0c8fa76867751046c89f69db5661ed4f/Lib/re/__init__.py#L226
        - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L30
- Call accept_charsets(self) function (line 389)
    - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L389
    - accept_charsets(self) function will get the list of accept charsets from this client from header
    - Call parse_accept_header() function (line 471)
        - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L471
        - parse_accept_header() function will parses an HTTP Accept-* header.
        - “_accept_re” is doing the actual parsing using the Compile() function, which will compile a regular expression pattern, returning a Pattern object.
            - Link: https://github.com/python/cpython/blob/5ea052bb0c8fa76867751046c89f69db5661ed4f/Lib/re/__init__.py#L

226

      - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L30
- Call accept_encodings(self) function (line 396)
  - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L396
  - accept_encodings() function will get the list of encodings this client accepts from the header.
  - Call parse_accept_header() function (line 471)
    - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L471
    - parse_accept_header() function will parses an HTTP Accept-* header.
    - "_accept_re" is doing the actual parsing using the Compile() function, which will compile a regular expression pattern, returning a Pattern object.
      - Link: https://github.com/python/cpython/blob/5ea052bb0c8fa76867751046c89f69db5661ed4f/Lib/re/__init__.py#L226
    - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L30
- Call accept_languages(self) function (line 404)
  - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L404
  - accept_languages() function will get list of languages this client accepts from the header
  - Call parse_accept_header() function (line 471)
    - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L471
    - parse_accept_header() function will parses an HTTP Accept-* header.
    - _accept_re is doing the actual parsing using the Compile() function, which will compile a regular expression pattern, returning a Pattern object.
      - Link:

https://github.com/python/cpython/blob/5ea052bb0c8fa76867751046c89f69db5661ed4f/Lib/re/__init__.py#L226

- Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L30

- Call cache_control(self) function (line 417)
  - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L417
  - Cache_control will get incoming cache control headers
  - Call parse_cache_control_header() function (line 524)
    - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L524
    - parse_cache_control_header() will parse a cache control header.
    - Call parse_dict_header() function (line 337)
      - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L337
      - parse_dict_header() function will parse lists of key, value pairs from header
        - Call parse_http_list() function (line 1431)
          - Link: https://github.com/python/cpython/blob/5ea052bb0c8fa76867751046c89f69db5661ed4f/Lib/urllib/request.py#L1431
          - parse_http_list() function will do the actual parsing work, parse comma-separated lists where the elements of the list may include quoted-strings.
        - Call unquote_header_value() function (line216)
          - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L216
          - unquote_header_value() function will remove quotes from a header value
- Call if_match(self) function (line 425)

  - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L425
  - if_match() function will get all the etags in the “If-Match” header
  - Call parse_etags function() (line 880)
    - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L880
    - parse_etags function() will parse an etag header.
    - “_etag_re” is doing the actual parsing using the Compile() function, which will compile a regular expression pattern, returning a Pattern object.
      - Link: https://github.com/python/cpython/blob/5ea052bb0c8fa76867751046c89f69db5661ed4f/Lib/re/__init__.py#L226
    - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L52
- Call if_none_match() function (line 433)
  - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L433
  - if_none_match() function will get all the etags in the “If-None-Match” header
  - Call parse_etags function() (line 880)
    - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L880
    - parse_etags function() will parse an etag header.
    - “_etag_re” is doing the actual parsing using the Compile() function, which will compile a regular expression pattern, returning a Pattern object.
      - Link: https://github.com/python/cpython/blob/5ea052bb0c8fa76867751046c89f69db5661ed4f/Lib/re/__init__.py#L226
    - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L52
- Call if_modified_since(self) (line 441)
  - Link:

https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L441

- - if_modified_since() function will get the parsed 'If-Modified-Since' header.
  - Call parse_date() function (line 918)
    - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L918
    - parse_date() function will Parse an :rfc:`2822` date into a timezone-aware
      - Call parsedate_to_datetime() function to convert the parsed date to datetime (line 197)
        - Link: https://github.com/python/cpython/blob/5ea052bb0c8fa76867751046c89f69db5661ed4f/Lib/email/utils.py#L197
- Call if_unmodified_since(self) function (line 450)
  - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L450
  - if_unmodified_since(self) will get the parsed ‘If-Unmodified-Since’ header.
  - Call parse_date() function (line 918)
    - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L918
    - parse_date() function will Parse an :rfc:`2822` date into a timezone-aware
      - Call parsedate_to_datetime() function to convert the parsed date to datetime (line 197)
        - Link: https://github.com/python/cpython/blob/5ea052bb0c8fa76867751046c89f69db5661ed4f/Lib/email/utils.py#L197
- Call if_range(self) function (line 459)
  - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L459
  - if_range(self) function will get the parsed 'If-Range' header.
  - Call parse_if_range_header() function (line 710)
    - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f

5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L710

    - parse_if_range_header() function will Parses an if-range header which can be an etag or a date.
    - Call parse_date() function (line 918)
      - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L918
      - parse_date() function will Parse an :rfc:`2822` date into a timezone-aware
        - Call parsedate_to_datetime() function to convert the parsed date to datetime (line 197)
          - Link: https://github.com/python/cpython/blob/5ea052bb0c8fa76867751046c89f69db5661ed4f/Lib/email/utils.py#L197
    - Call unquote_etag() function (line 855)
      - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L855
      - unquote_etag() function will Unquote a single etag.
- Call range(self) function (line 470)
  - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L470
  - range() function will get the parsed `Range` header.
  - Call parse_range_header() function (line 728)
    - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L728
    - parse_range_header() function will Parses a 'range' header.
- Call user_agent(self) function (line 482)
  - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L482
  - user_agent() function will get the ‘User-Agent’ header value.
    - Call user_agent_class() (line 97)
      - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/sansio/request.py#L97

- user_agent_class() is a ‘UserAgent’ type
  - class UserAgent represents a parsed user agent header value.

- Call authorization(self) function (line 498)
  - Link: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/request.py#L498
  - authorization() function will get ‘Authorization’ header value
    - Call parse_authorization_header() function (line 641)
      - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L641
      - parse_authorization_header() function will Parse an HTTP basic/digest authorization header transmitted by the web browser.
      - If auth_type == ‘digest’
        - Call parse_dict_header() function (line 337)
          - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py#L337
          - parse_dict_header() function will parse lists of key, value pairs from header
            - Call parse_http_list() function (line 1431)
              - Link: https://github.com/python/cpython/blob/5ea052bb0c8fa76867751046c89f69db5661ed4f/Lib/urllib/request.py#L1431
              - parse_http_list() function will do the actual parsing work, parse comma-separated lists where the elements of the list may include quoted-strings.
            - Call unquote_header_value() function (line216)
              - Link: https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a

/src/werkzeug/http.py#L216

- unquote_header_value() function will remove quotes from a header value