Open-Source Report

Proof of knowing your stuff in CSE312

Guidelines

Provided below is a template you must use to write your reports for your project.

Here are some things to note when working on your report, specifically about the **General Information & Licensing** section for each technology.

- **Code Repository**: Please link the code and not the documentation. If you'd like to refer to the documentation in the **Magic** section, you're more than welcome to, but we need to see the code you're referring to as well.
- **License Type**: Three letter acronym is fine.
- **License Description**: No need for the entire license here, just what separates it from the rest.
- **License Restrictions**: What can you *not* do as a result of using this technology in your project? Some licenses prevent you from using the project for commercial use, for example.

Also, feel free to extend the cell of any section if you feel you need more room.

If there's anything we can clarify, please don't hesitate to reach out! You can reach us using the methods outlined on the course website or see us during our office hours.

Flask

General Information & Licensing

Code Repository	https://github.com/Ai-Jesse/Blue-Jesse
License Type	MIT
License Description	• Any user can obtain this software free of charge • Any user is allowed to copy, modify, and distribute • Users are allowed to sell the software
License Restrictions	• The software is provided “as is” without warranty • The authors or copyright holders can't be liable for any claim, damages, or other liability

Magic ⋆★˚｡⋆☽˚⁀➴｡˚★⋆

Parsing HTTP Headers using flask:

Flask uses many libraries to parse the http headers, the steps are:

After establishing TCP connection using socketserver, when a request is received, it will create a BaseHTTPRequestHandler from http library that extends StreamRequestHandler from socketserver which also extends BaseRequestHandler which is what deals with all of the requests recieved.

BaseRequestHandler: https://github.com/python/cpython/blob/main/Lib/socketserver.py line 723

StreamRequestHandler: https://github.com/python/cpython/blob/main/Lib/socketserver.py line 769

BaseHTTPRequestHandler: https://github.com/python/cpython/blob/main/Lib/http/server.py line 146

In BaseHTTPRequestHandler, we called the parse_request method which is used to parse the request including the header. To parse the header, it pass the request to the parse_headers method which also uses the _read_headers to parse the request header by going through lines in the request. The header is then stored as a list in the headers attribute in BaseHTTPRequestsHandler.

parse_request: https://github.com/python/cpython/blob/main/Lib/http/server.py line 267

parse_headers: https://github.com/python/cpython/blob/main/Lib/http/client.py line 224

_read_headers: https://github.com/python/cpython/blob/main/Lib/http/client.py line 206

Afterward, flask also uses the wekzeug library, using the WSGIRequestHandler class in wekzeug library that also extends BaseHTTPRequestsHandler, it call the make_envrion method which takes the headers attribute in BaseHTTPRequestsHandler and parse it into a dictionary format and store it in an local environ variable which then get returned and set as an attribute called environ in the WSGIRequestHandler class in the run_wsgi method.

WSGIRequestHandler: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/serving.py line 148

make_environ: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/serving.py line 159

run_wsgi: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/serving.py line 239

To be more specific on above, when a request is received, it calls the handle method from the serving class in werkzeug library whic also extends the server class in http, which then calls the handle_one_request method and calls run_wsgi method.

werkzeug handle: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/serving.py line 360

http handle: https://github.com/python/cpython/blob/main/Lib/http/server.py line 427 handle_one_request: https://github.com/python/cpython/blob/main/Lib/http/server.py line 390

When run_wsgi is called, it also define and call a local method called execute which calls the __call__ method in app class of flask and pass it the environ variable that contains the header, __call__ then call the wsgi_app method and pass the environ variable, then this calls the request_context method and passes the environ variable, then this returns a RequestContext object from flask library, which creates a request object from werkzeug library that takes in environ as a constructor.

execute: https://github.com/pallets/werkzeug/blob/main/src/werkzeug/serving.py line 319

__call__: https://github.com/pallets/flask/blob/main/src/flask/app.py line 2546

wsgi_app: https://github.com/pallets/flask/blob/main/src/flask/app.py line 2495

RequestContext: https://github.com/pallets/flask/blob/main/src/flask/app.py line 2423

request: https://github.com/pallets/flask/blob/main/src/flask/ctx.py line 278

Then finally flask have a request wrapper object that extends the request object from werkzeug created above which we can then use in our server by importing request from flask and use flask.request which gives us the dictionary of headers that was parsed and saved in the environ created before.

request: https://github.com/pallets/flask/blob/main/src/flask/wrappers.py line 15