TCP Connections Open-Source Report

Proof of knowing your stuff in CSE312

Flask (Python)

General Information & Licensing

Code Repository	https://github.com/pallets/flask
License Type	BSD 3-Clause New or Revised License
License Description	This type of license is a permissive free software license similar to BSD 2-Clause with minimal limitations. Here, an additional clause exists prohibiting users from using the copyright holder's name to promote derived products without written consent. Drawing directly from the flask license: “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.” The following are permissible under the above license: 1. Commercial Use 2. Modification 3. Distribution 4. Private Use
License Restrictions	Following from the above, this license allows for a wide range of permitted use. Restrictions, however, do exist surrounding liability and warranty. Copyright holders have explicitly distanced themselves from association with the products derived from this framework without explicit written consent.

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.

*This section will likely grow beyond the page

- In class we discussed how to parse the http request into ints proper request type, and based on the requirement we would use for different task. In our group project we used the flask header parser to make it easy on us for parsing and getting the header.

```
class Flask(Scaffold):
```

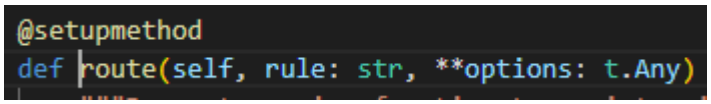
- This object is created and it acts as a central registry that stores the url rules, templates, configuration. The “Scaffold” is the parameter that is the package, which would be resolved based on the python code for the package.

```
flask.scaffold.Scaffold
@setupmethod
def route(self,
          rule: str,
          **options: Any) -> (T_route) -> T_route
```

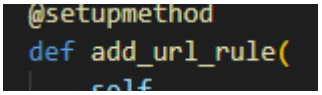
- This calls flask folder then the package folder inside that we specific function. Within the route function we specide the “rules” which is how we will implement the url.

```
@app.route('/')
```

route():

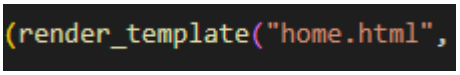
- route(‘/’) function within the scaffold class in flask/src/flask/scaffold.py , line 423 takes in the added URL as an input. An extra method can be added to the function to be executed once the URL is received.
- 

- https://github.com/pallets/flask/blob/cc66213e579d6b35d9951c21b685d0078f373c44/src/flask/scaffold.py#L423
- The route is connected to the scaffold.py folder, within that file there is method called setpumethod, which calls the add_url_rule.

- 

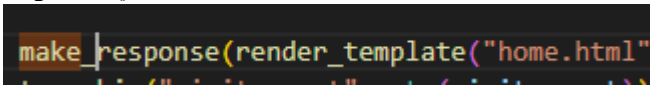
- https://github.com/pallets/flask/blob/cc66213e579d6b35d9951c21b685d0078f373c44/src/flask/scaffold.py#L455
- The function takes incoming routing incoming request and building urls.using. It uses the “route” decorator to view_func. If the endpoint isnt then it will go to the default name. However, if the name already exists then an error will be raised.

render_template():

- 

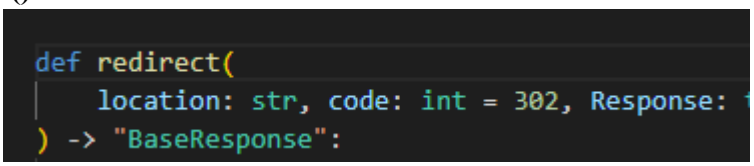
- https://github.com/pallets/flask/blob/cc66213e579d6b35d9951c21b685d0078f373c44/src/flask/templating.py#L135
- This method causes flask to look for the templates folder and searches for the different file names. But it takes to the *templating.py* which calls the function *_render()* which takes the app and the template. If a list of template is given, then the first name that exists will be rendered. And the context variable makes it available in the template.

make_response():

- 

- https://github.com/pallets/flask/blob/cc66213e579d6b35d9951c21b685d0078f373c44/src/flask/helpers.py#L163
- The make _response takes to the helpers.py files, it returns a response object which can be use to attach to the header.

redirect():

- 

- https://github.com/pallets/flask/blob/cc66213e579d6b35d9951c21b685d0078f373c44/src/flask/helpers.py#L266
- The redicte() function is suede to redirect to another local. Depending the the type we are directing the function does different things
- If it is an application then in will be redirect() return statement.
- If anything else it will use **`_wz_redirect()`** and it pass a type of 300 response code, with except of 300 response as it is not considered a real redirect. And it doesnt have 304 as it is the answer for a request with a request with defined If-Modified-Since headers.

run():

- ```
  if __name__ == '__main__':
      app.run(debug=True, host='0.0.0.0', port='5000')
  ```
- https://github.com/pallets/flask/blob/cc66213e579d6b35d9951c21b685d0078f373c44/src/flask/app.py#L1191
- We implement flask run() which runs with debug mode true so we can see the debugger in case of exception.

https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/wrappers/request.py

- Defines request class in wekzeug

https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/datastructures.py#L1354

- Starting at 1354, data structure of headers environment is defined

Final Destination:
https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/http.py

- Code parses headers based on format set by environment

-
  ```