

TCP Connections 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 / Python

General Information & Licensing

Code Repository	https://github.com/pallets/flask
License Type	BSD 3-Clause "New" or "Revised" License
License Description	<ul style="list-style-type: none">• A permissive license similar to the BSD 2-Clause License, but with a 3rd clause that prohibits others from using the name of the copyright holder or its contributors to promote derived products without written consent
License Restrictions	<ul style="list-style-type: none">• Liability• Warranty

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

Flask is a web application framework written in Python and depends on the Jinja template engine and the Werkzeug WSGI toolkit. Flask listens to a port on TCP socket and receives an HTTP requests and parses the data. It parses it based on the protocol format and makes it available for users in a convenient manner.

Flask:

Wrappers.py: where the request class is located

Views.py: where parsing the header request method is

located(<https://github.com/pallets/flask/blob/cc66213e579d6b35d9951c21b685d0078f373c44/src/flask/views.py#L135>)

Testing.py: possibly where all the params are created?

Start:

- `app = Flask(__name__)` (our server code)
 - Ctrl click flask to go into flask library
 - <https://github.com/pallets/flask/blob/066a35dd322f689ec07d7c0e82b19eac-adac3c6b/src/flask/app.py#L110>
- `run()`
 - Ctrl f for run since the description for run is `"""Runs the application on a local development server."""`
 - <https://github.com/pallets/flask/blob/066a35dd322f689ec07d7c0e82b19eac-adac3c6b/src/flask/app.py#L1077>
- `run()` comment description
 - In the run description, werkzeug can be found as one of the param options
 - <https://github.com/pallets/flask/blob/066a35dd322f689ec07d7c0e82b19eac-adac3c6b/src/flask/app.py#L1115>
 - `server. See :func:`werkzeug.serving.run_simple` for more information.`
 - Ctrl-f and find `run_simple` function
- `Run_simple`
 - <https://github.com/pallets/flask/blob/066a35dd322f689ec07d7c0e82b19eac-adac3c6b/src/flask/app.py#L1191>
 - Ctrl click on `run_simple` leads to werkzeug library

- <https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/serving.py#L907>
- In the parameter description it says “Start a development server for a WSGI application. Various optional features can be enabled.”
- <https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/serving.py#L925>
- Scroll down to see where the server is being made
- `Srv = make_server()`
- Ctrl click `make_server`
- **Make_server**
 - <https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/serving.py#L853>
 - Leads to `src/werkzeug/serving.py`
 - <https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/serving.py#L864>
 - Description is “Create an appropriate WSGI server instance based on the value of `threaded` and `processes`.”
 - Scroll down to return `BaseWSGIServer`
 - Ctrl click on `BaseWSGIServer`
 - <https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/serving.py#L893>
- **BaseWSGIServer**
 - <https://github.com/pallets/werkzeug/blob/3115aa6a6276939f5fd6efa46282e0256ff21f1a/src/werkzeug/serving.py#L651>
 - Ctrl click on `HTTPServer`
 - Leads to class `HTTPServer(socketserver.TCPServer):`
 - <https://github.com/python/cpython/blob/5ea052bb0c8fa76867751046c89f69db5661ed4f/Lib/http/server.py#L129>
- **TCP connections**
 - Connection is made is made at line 135
 - <https://github.com/python/cpython/blob/5ea052bb0c8fa76867751046c89f69db5661ed4f/Lib/http/server.py#L135>
 - Host and port are created at line 136
 - <https://github.com/python/cpython/blob/5ea052bb0c8fa76867751046c89f69db5661ed4f/Lib/http/server.py#L136>
- What does this technology (library/framework/service) accomplish for you?
TCP helps us secure the transmissions between our server and each client. TCP ensures that the data is properly transmitted and received and provides error-checking mechanisms to ensure the integrity of the data being transmitted.
- Explain what this technology does in your project. What problems does it solve for you?
TCP establishes a stable connection between the server and each connected client. Since it uses error checking mechanisms and retransmission mechanisms, the data delivered is accurate and without errors for sure. So when a player logs in with credentials unique to only that player, TCP ensures that the information being displayed and sent to is for that user only and not another user.

TCP also ensures that relevant data is delivered in a relevant order so that each person/client gets data in proper order and it's easier for the client to reassemble the data back into the original message. When a player accesses their player profile or leaderboard, the information is sent in a proper order so that when the player decodes it and reassembles it the information is legible and correct.

TCP also handles flow of data, so it regulates how much data is being sent and if there is a possibility for overload it reduces the amount of data according to the receiver's capability. For example: If multiple requests are made and huge chunks of data has to be sent to the user, if the user cannot handle the entire amount of data at once, TCP will make sure only the minimum amount is sent so as to not overload the user and slow down the connection.

Overall, TCP delivers data in a reliable and efficient manner.