Open-Source Technology Use Report

Proof of knowing your stuff in CSE312

Guidelines

Provided below is a template you must use to write your report for each of the technologies you use in 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'd like 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.
- **Who worked with this?**: It's not necessary for the entire team to work with every technology used, but we'd like to know who worked with what.

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-Socket.io]

General Information & Licensing

Code Repository	https://github.com/miguelgrinberg/Flask-SocketIO
License Type	MIT License
License Description	• Commercial use • Modification • Distribution • Private use
License Restrictions	• Liability • Warranty
Who worked with this?	Zaki, Kyle, Tenzin, Gorden

Use as many of the sections below as needed, or create more, to explain every function, method, class, or object type you used from this library/framework.

SocketIO

Purpose

- This object is used to start up the server and allow for messages to be both received and sent from the server. Through this object we can use the functions emit() and on() where emit sends a message to all connected listeners for that message while on listens and receives messages to all connected emitters. We also use run() to start up the server
- It's used like:
    - For emit() it usually sends the event to all listeners on the namespace due to broadcast equalling true and sometimes sending parameters for the listening function to use.
    - For run() we use it once to run the server with the flask, the port and host and our optional option debug =true
    - For on, with different messages we were able to receive data to store on our server which allows us to share between the sender and all the other clients/listeners.
      https://github.com/TenzinR/CSE312-GroupWebsite/blob/main/app.py#L155
      For example, line 155, when on receives message "connect" we are able to validate the user, then check if they are in our online database, if not we add them to it and add them to the online list, which would update every other client's page and tell them that user has come online.

Magic

https://github.com/miguelgrinberg/Flask-SocketIO/blob/df04df439535ad5eb7ce910ae1e8204aed3cabfc/src/flask_socketio/__init__.py#L54 line 54
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/__init__.py Line 516

- Contains the run function which sets the host if it has a value which we gave localhost and a port which we gave as well at 8000 and since we used no other parameters we can go to line 629 with run_server() which runs the server

https://github.com/miguelgrinberg/Flask-SocketIO/blob/df04df439535ad5eb7ce910ae1e8204aed3cabfc/src/flask_socketio/__init__.py#L401 line 401

- This emit function is used when we want to send messages to all connected listeners.
- To begin it takes the parameters called event which is the string of the event it wants to send out to listeners, arg's which is a dictionary with json data and in our example since we use broadcast = true for our parameters in emit(), we send the message to all listeners on the namespace because the namespace defaults to global.
- Line 436 sets the namespace to default for us since we leave it empty
- We skip lines 440-459 since those values are set to none and since this is a wrapper function we can go into the actual emit function
- The emit function on line 782 essentially returns a socketio.emit which sends a server message to all connected listeners that contains all the data that was either set to the default value of none or the values that we registered such as the message which is the string that the listeners are waiting for, the data and/or the broadcast=true, which sends it to all listeners connected to the global namespace.

https://github.com/miguelgrinberg/Flask-SocketIO/blob/df04df439535ad5eb7ce910ae1e8204aed3cabfc/src/flask_socketio/__init__.py#L258 line 258

- The on function must be used with a handler. Which decorates the handler so it has additional features.
- On takes a message param. When it receives that message, its handler that it decorated/wrapped takes over.
- Name_space is automatically "/" since we didn't initialize it.
- The event names message, json, connect and disconnect are reserved already and cannot be used for other named events.
- On line 282, it calls handle_event which if it doesn't recognize the client, it does not handle the event.
- When it does, it checks manage_session. Because we passed SocketIO the flask application, manage_session is true, and thus it passes back the handler the arguments/message to do whatever we implemented in our handler.