CSE 312 Project

Ishaan Roychowdhury, Dannen Roberts, Manthan Vasavada

Flask-SocketIO Report

For this project, when dealing with web sockets, we used two libraries, one on the front end and one on the backend. The idea of this was that when going to another tab, and loading our site on that other site, the content displays in real-time on both tabs almost instantaneously. This required the usage of web sockets. To fulfill the backend requirements of this we used Flask-WebSocketIO.

Similar to our homework, this framework was responsible for opening and closing WebSockets, in which we could send and receive information really quickly. The idea in this tool is that multiple processes are put into a queue, in which the information can be updated for multiple sockets pretty quickly. It has the ability to broadcast messages as well. When a message is sent with the broadcast option every socket connected to the namespace receives that message. Since broadcasting is made easy, we are able to send to the sockets that we create through the framework quickly, and the information will update on multiple threads, or tabs if needed. When each socket receives a message, it is considered an event. These events need to be handled by the server with a handler, in the same way, that routes are handled by functions in Flask. This framework provides the server-side functionality to send, receive and broadcast messages quickly. This means that we can ensure that sockets all have the same information quickly and asynchronously. This framework takes care of all of the things that we had to worry about in homework 6 like setting up a multithreaded system and storing all of the web sockets in some sort of data structure.

We added functionality for chat rooms and DM's through the functions in the flask_socket_io library called rooms. This helps send messages to specific users securely to one another. It allows a user to join and leave rooms based on the session ID of that connection. The objective of the Sprint where we had to deal with messages was accomplished through this. We made necessary changes in our homepage.py file which handles messaging. This bi-directional communication is established between web clients and servers on our website. One can send messages through the emit function, which takes in the session ID of the client as it's room argument.

When it comes to licensing, this tool is under the MIT license. You are allowed to use it in any way you see fit. According to the MIT license, a common open source license, you are allowed to work with modify and distribute the code in any way you see fit. The creators are not responsible or liable, for your code in any way.

A Couple of Links to the Flask SocketIO Code and Documentation Link to the GitHub:

https://github.com/miguelgrinberg/Flask-SocketIO Link to where some of the Sockets are handled:

https://github.com/miguelgrinberg/Flask-SocketIO/blob/master/flask_socketio/__init__.py

Link to where the Namespace is handled:

https://github.com/miguelgrinberg/Flask-SocketIO/blob/master/flask_socketio/namespace.py

Flask SocketIO Documentation Link:

https://flask-socketio.readthedocs.io/en/latest/

Link to the flask_socket_io Rooms example we found helpful →

https://github.com/miguelgrinberg/Flask-SocketIO/blob/master/example/app.py

More links for documentation on rooms →

flask-socketio.readthedocs.io/en/latest/