Open-Source Report: Websockets

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.

Python Flask

General Information & Licensing

Code Repository	https://github.com/miguelgrinberg/Flask-SocketIO https://github.com/miguelgrinberg/python-socketio
License Type	MIT
License Description	• Modification: The license permits modification of the software, allowing users to create derivative works. However, it requires that any modified files must include the original copyright notice and disclaimer.

	• Use: The license grants users the right to use the software for any purpose, including private and commercial use.
License Restrictions	• Redistribution of source code must retain the copyright notice • The author or copyright holder shall not be liable for any claims • There is no warranty given to users.

Magic

__init__(self, app=None, **kwargs) create a SocketIO object to get the Flask app instance (standard initialization), which will be used later. Line 171
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/__init__.py

init_app(self, app, **kwargs) is used to create and handle all Socket connections. Line 191
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/__init__.py

emit(self, event, *args, **kwargs) can emit a SocketIO event to one or more connected clients. This function can be used outside of a SocketIO event context. Line 401
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/__init__.py

emit(event, *args, **kwargs) can emit a SocketIO event to one or more connected clients. Line 847
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/__init__.py

connect(self, namespace=None, query_string=None, headers=None, auth=None) is used to connect clients. The connection is automatically established when an instance of this class is created. Line 91
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/test_client.py

Packet.py file can create packets to transmit between websocket frames. The encode(self) can encode the packet for transmission (Line 45), and the decode(self) can decode a transmitted package(Line 69).
https://github.com/miguelgrinberg/python-socketio/blob/main/src/socketio/packet.py

Class server implements a fully compliant Socket.IO web server with support for websocket and long-polling transports. Line 13
https://github.com/miguelgrinberg/python-socketio/blob/main/src/socketio/server.py

emit(self, event, data=None, to=None, room=None, skip_sid=None, namespace=None, callback=None, ignore_queue=False) can emit a custom event to one or more connected clients. Line 271
https://github.com/miguelgrinberg/python-socketio/blob/main/src/socketio/server.py

get_session(self, sid, namespace=None) can return the user session for a client. Line 477
https://github.com/miguelgrinberg/python-socketio/blob/main/src/socketio/server.py

session(self, sid, namespace=None) can return the user session dictionary for the client. Line 507
https://github.com/miguelgrinberg/python-socketio/blob/main/src/socketio/server.py

Class Client(object) implements a fully compliant Socket.IO web client with support for websocket and long-polling transports. Line 35
https://github.com/miguelgrinberg/python-socketio/blob/main/src/socketio/client.py