

WebSocket Open-Source 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 SocketIO

General Information & Licensing

Code Repository	https://github.com/fcap25/CSE312-Project
License Type	MIT
License Description	<ul style="list-style-type: none">• License is free of charge and without restriction• License can be used without limitation (can be sold)• No warranty or liability provided
License Restrictions	<ul style="list-style-type: none">• The copyright license must be issued with usage of the software

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 WebSocket is created
- Socket.IO is built on top of the websocket protocol, but provides additional fallbacks such as long-polling or additional reconnects. Due to these extra features, Socket.IO has different metadata from a plain websocket connection, and thus a plain websocket client cannot connect to a Socket.IO server (and vice versa). After the connection is created, Socket.IO provides extra features that allow it to be more reliable than plain websockets. For example, packet buffering and automatic reconnection provide more stability during the connection as well as during any issues that might arise (packet loss, broken pipes).
- 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.

socket.run() runs the SocketIO web server:

https://github.com/miguelgrinberg/Flask-SocketIO/blob/326cef7c55c71cb16c506a30da60abcbacd59871/src/flask_socketio/ init .py#L553

@socketio.on() is the wrapper function for handling communication over the socket:

https://github.com/miguelgrinberg/Flask-SocketIO/blob/326cef7c55c71cb16c506a30da60abcbacd59871/src/flask_socketio/ init .py#L258

handle_request() is called from socket.on() to handle the message of the request:

https://github.com/miguelgrinberg/Flask-SocketIO/blob/326cef7c55c71cb16c506a30da60abcbacd59871/src/flask_socketio/_init_.py#L789

`emit()` is called from `socketio.on` and is used to send data over the socket connection:

https://github.com/miguelgrinberg/Flask-SocketIO/blob/326cef7c55c71cb16c506a30da60abcbacd59871/src/flask_socketio/ init .py#L847

*This section will likely grow beyond the page

