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.

# websockets

## General Information & Licensing

|  |  |
| --- | --- |
| Code Repository | https://github.com/aaugustin/websockets/ |
| License Type | [BSD 3-Clause "New" or "Revised" License](https://github.com/aaugustin/websockets/blob/main/LICENSE) |
| License Description | * Private/Commercial Use Permitted * Modification Permitted * Distribution Permitted |
| License Restrictions | * Creators are not liable for damages code may produce * No warranty is provided * License/Copyright notice must be provided in licensed material |
| Who worked with this? | Colin |

*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.*

# websocket.serve

## Purpose

|  |
| --- |
| * Creates the websocket server at a given IP address/port, and sends incoming messages to a customizable callback function * This will be used to initialize the websocket server within websocket.py of our repo (line number tbd) |

## *Magic* ⋆★͎۪۫｡˚۰˚☽˚⁀➷｡˚★彡͎۪۫⋆ ༄

|  |
| --- |
| * an instance of [serve](https://github.com/aaugustin/websockets/blob/13eff12/src/websockets/legacy/server.py#L826-L1048) is created by calling websockets.serve. * Serve creates a [WebSocketServer](https://github.com/aaugustin/websockets/blob/13eff12/src/websockets/legacy/server.py#L640-L823), which creates, holds, and destroys WebSocketServerProtocols. * Serve also creates a [WebSocketServerProtocol](https://github.com/aaugustin/websockets/blob/13eff12/src/websockets/legacy/server.py#L61-L637), which receives, processes, and sends out data to the server * The handler for WebSocketServerProtocol waits for a handshake to commence, holding the entire handshake process in the [handshake](https://github.com/aaugustin/websockets/blob/13eff12/src/websockets/legacy/server.py#L548-L637) function * The protocol created waits for an attempt at a valid handshake request, which is determined by reading the raw bytes and attempting to convert into a HTTP request using the [read\_http\_request](https://github.com/aaugustin/websockets/blob/13eff12/src/websockets/legacy/server.py#L254-L281) function. * The proper HTTP handshake response is validated and processed by handshake, which after reading the HTTP request builds the response using [build\_response](https://github.com/aaugustin/websockets/blob/13eff12bb4c995b50154fdc250281c92ddccaca0/src/websockets/legacy/handshake.py#L106-L120) and writes it back to the client through [write\_http\_response](https://github.com/aaugustin/websockets/blob/13eff12bb4c995b50154fdc250281c92ddccaca0/src/websockets/legacy/server.py#L283-L309). |

# websocket.send

## Purpose

|  |
| --- |
| On a pre-existing websocket, send takes a string or bytes, formats it into a websocket frame, and then sends it to the client. |

## *Magic* ⋆★͎۪۫｡˚۰˚☽˚⁀➷｡˚★彡͎۪۫⋆ ༄

|  |
| --- |
| * The [send](https://github.com/aaugustin/websockets/blob/13eff12bb4c995b50154fdc250281c92ddccaca0/src/websockets/legacy/protocol.py#L564-L722) coroutine (which WebSocketServerProtocol inherits from WebSocketCommonProtocol) is called with something that can be used as a payload (string, bytes) * The payload is given to [write\_frame](https://github.com/aaugustin/websockets/blob/13eff12bb4c995b50154fdc250281c92ddccaca0/src/websockets/legacy/protocol.py#L1179), who in turn calls write\_frame\_sync * [write\_frame\_sync](https://github.com/aaugustin/websockets/blob/13eff12bb4c995b50154fdc250281c92ddccaca0/src/websockets/legacy/protocol.py" \l "L1154-L1162) creates an instance of [Frame](https://github.com/aaugustin/websockets/blob/13eff12bb4c995b50154fdc250281c92ddccaca0/src/websockets/legacy/framing.py), which through instantiation and it’s own write function constructions the frame to write, and then writes to the TCP |

# websocket.broadcast

## Purpose

|  |
| --- |
| Allows us to send a websocket frame to multiple websocket connections simultaneously. |

## *Magic* ⋆★͎۪۫｡˚۰˚☽˚⁀➷｡˚★彡͎۪۫⋆ ༄

|  |
| --- |
| Operates fairly similar to websocket.send. Essentially [broadcast](https://github.com/aaugustin/websockets/blob/13eff12/src/websockets/legacy/protocol.py#L1547-L1597) takes a list of websockets and data, and for each websocket in the list calls its write\_frame\_sync. |

# [insert method/function/class/object name here]

## Purpose

|  |
| --- |
| Replace this text with some that answers the following questions for the above tech:   * What does this tech do for you in your project? * Where specifically is this tech used in your project? Give us some details like file location and line number, if applicable. If too cumbersome, a general description of where it’s used for a given purpose is fine as well. |

## *Magic* ⋆★͎۪۫｡˚۰˚☽˚⁀➷｡˚★彡͎۪۫⋆ ༄

|  |
| --- |
| 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 for you in the **Purpose** section of this report? Please explain this in detail, starting from after the TCP socket is created. Remember, to be allowed to use a technology in your project, you must be able to know how it works. * 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 may grow beyond the page for many features. |