

For Future Developers

File Architecture and Important Files

The main files appear as follows:

```
Server
├── index.py
├── dbopsAttempt.py
├── mlops.py
├── requirements.txt
├── nltk.txt
├── Procfile
├── templates
│   ├── index.html
│   └── chats.html
└── static
    └── Image Files
```

index.py

index.py is the main server code. The bulk of it dedicated to the background of the server setup and the socketIO event-handling code. While many things in each event can be changed while keeping the code modular, it is important that the socketIO interactions are either unchanged, or changed in both front/backend.

index.html

index.html is the main chat page. It sends, receives, and renders messages, as well as prepping links and images with the appropriate HTML tags. This is the other end of most of the SocketIO emits/ons in `index.py`.

dbopsAttempt.py

This is where all of the database code is handled. After it is initialized with a database engine by flask, it handles all the direct communication with the database.

mlops.py

This file contains all of the code for the subject analysis component of Tiger Message. It contains code to pull out objects and subjects in sentences in addition to code which takes lists of these and calculates a “closeness” between 0 and 1 using WordNet.

chats.html

This contains the “about” page and the list of a user’s chats. There is one socketio event parsed by this site, which simply refreshes the chat list.

Other Files

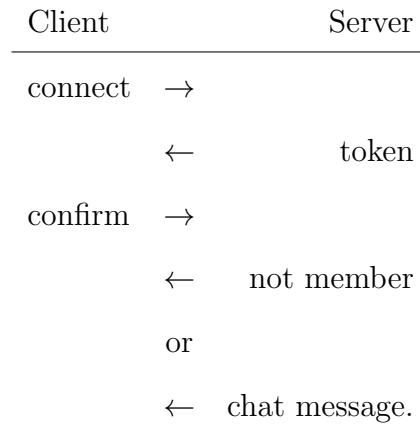
The other files highlighted above are dependency information and run commands for Heroku to correctly parse and run the server.

Key SocketIO Workflows

There are several socketIO events passed back and forth between the server and the client

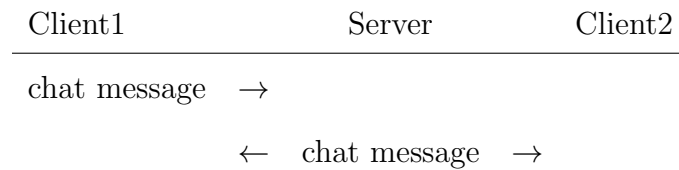
User Connects

When a user connects, they go through the following process, exchanging user information back and forth with the server.



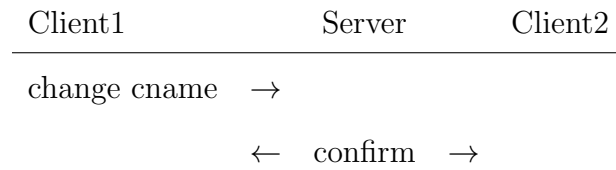
User Sends a Message

If client1 sends a message in a chat with client2, the message is processed by the server, and then distributed to all users in the selected chat.



Changing a Chat's Name

If client1 changes the name of a shared chat with client2, the change is processed by the server and then confirmation to re-check the chat's name is sent to each website.



Single Event Interactions

For certain things like restricting the messages to a subject, re-displaying all chat messages, reloading pages, there is just a single event that the server emits to all the necessary pages.