Folder Overview

**index.html <- Frontend | Backend -> pubsub.js**

index.html serves as the console for Publishing of Data from different agencies to help to simulate the flow of data and how it works.

pubsub.js allows different agencies to connect to Solace network to publish and subscribe from the console.

**login.html <- Frontend | Backend -> login.js**

login.html provides the interface that is accessed by agencies and the admin to access their relevant pages.

Login.js assists in authenticating the login detail inputted (via semp call for agency users and hardcoded values for admin users)

**register.html <- Frontend | Backend -> processregister.js**

register.html provides the interface that new users can access to get information on solace and the solace software and cloud broker, they can also register for access via the form available here

processregister.js assists in publishing the registrations details to a que, to be consumed by an admin who can choose to establish a bridge with this new users broker

**homepage.html**

Provides the interface that is accessed by agencies to access the various central broker functionalities that are available to a registered broker

**registertopiv.html <- Frontend | Backend -> processtopicreg.js**

register.html provides the interface that users can access to register new topics that they wish to publish to in the central broker

processtopicreg.js assists in publishing the topic details to a que, to be consumed by an admin who can choose to accept or reject the topic and create the relevant ACL allowance

**Admin.html <- Frontend | Backend -> log.js**

Admin.html is the front-facing interface that will be utilized by the Admin to view message logs and accept or deny new topic requests and broker registrations

log.js assists to establish connection with the solace broker and consumes from the various queues and writes them into the appropriate table on Admin.html

**AccessListCall.js**

This file is a function to handle Access Control for every agency that is connected to the central broker. Agencies can only publish or subscribe based on verified permission from central broker.

**config.js**

Contains sensitive information as simulated NEA and LTA to connect to Solace Cloud and handle SEMP Protocol. The file should be hidden from public.

**BrokerRegistration.js**

Attempts the register any new broker on central broker with provided details. Handles creation of new ACL and client usernames, as well as the bridge between central and new broker.

**dataflow.html <- Frontend | Backend -> Centralpubsub.js**

dataflow attempt to provide the real-time data flow chart for each agency on the frontend that logins to the web page. On the other hand, the backend serves as the Central Analytics that processes data published from the console and republishes new topic with the processed data. When the chart starts “consuming messages”, the central analytics will start its processing from the queue.

Centralpubsub.js <- Main Backend | Sub Backend -> EventMsg.js

As the backend processes data, it triggers two events: updating the data flow and processing the data into analytics. EventMsg receives the message and attempts to republish the processed message.

**govtech.js & tempData.js**

Gets the API data from data.gov.sg when it’s being called. Used by the console to publish simulated data. (Reference to pubsub.js) Both js file does the same thing but applies for different dataset. tempData is meant for temperature dataset while govtech.js is more general.

**microsoftanalyse.js**

Calls the analytics result from Microsoft Cognitive API with the published traffic Images. Used by the central broker to process traffic images. (Reference to EventMsg.js)

**map.html <- Frontend | Backend -> pubsubmap.js, selectorFiltering.js**

Google Map demonstration of the published data through Solace network. Data is decoupled and works independently from one another. Pubsubmap.js in charge of receiving subscribed data while selectorFiltering.js attempts to display received data on map.

**test.js**

The diagram for introductory instruction. Used in homepage.html and index.html