General working

The index.js file sitting in the root directory is the backbone of the premium app. There are two ports that it listens on –

8080 is where the premium app and all its functionality (email to backend, etc.) is served.

8081 is where the account duplication related tasks are served.

**Port 8080 (Premium app) –** When the ‘subscribe’ button is clicked on the app, as it does the necessary configuration in the Genesys org, it also collects some information for our use and posts it to the **‘/mail’** route. This route receives the information sent, creates an email body with an approval URL that contains all the information received encoded as an approval token. The approval URL is a ‘get’ link connecting to the 8081 port on the script that serves the account creation page. So, when this link is clicked, the account creation page opens up pre-populated with the information sent in the link.

**Port 8081 (Account duplication) –** When the approval link is clicked, it accesses the **‘/form’** route. The main function of this route is to serve the html form prefilled with the available information. The html form being served has uniquely identified placeholder values (iname, iorgName, etc.) that can be replaced with the actual information before it is served. The random account password is also being generated here.

Once the form is filled and submitted, the information is posted to the **‘/duplicate’** route, where the account duplicator is invoked with the received values. The success of the duplication is judged based on the presence or absence of the ‘”message":"Account duplicated.”’ string in the output log. Depending on success or failure, appropriate emails are sent. In case of success – the customer and Sergio bot are emailed, in case of failure, Sergio bot and the marketplace maintainer.

**Logs –** Some information is being logged, each date in a separate file, which can be looked at if things don’t add up – The approval email send results (success/failure), the results of the account duplication operation, the emails triggered after the account duplication operation (success/failure).

How to run

The server can be run from the root directory using the command – node index.js. To access the app, open [**http://localhost:8080/wizard**](http://localhost:8080/wizard) on a browser.

Configuration and dependencies

One email account will be needed for all the emails. The ‘accelerator-genesyscloud@cyara.com’ can safely be used for this (It’s also where the billing app gets its information).

External Node libraries required – nodemailer, generate-password, winston, winston-daily-rotate-file, dotenv.

**Environment variables -** Before running the app, certain values need to be defined in the .env file in the root directory.

SENDER\_ID - email id to be used to send all the emails to the account approver, the customer, Sergio bot, etc.

SENDER\_APP\_PASSWORD – the app password for the sender email.

The rest of the values are also receiver email ids, and should be self-explanatory.

**App uninstall –** For further development, the app might need the uninstall button. This is currently commented out in the ‘\docs\premium-app-sample\index.html’ file. Uncommenting it will conjure the button in the UI.