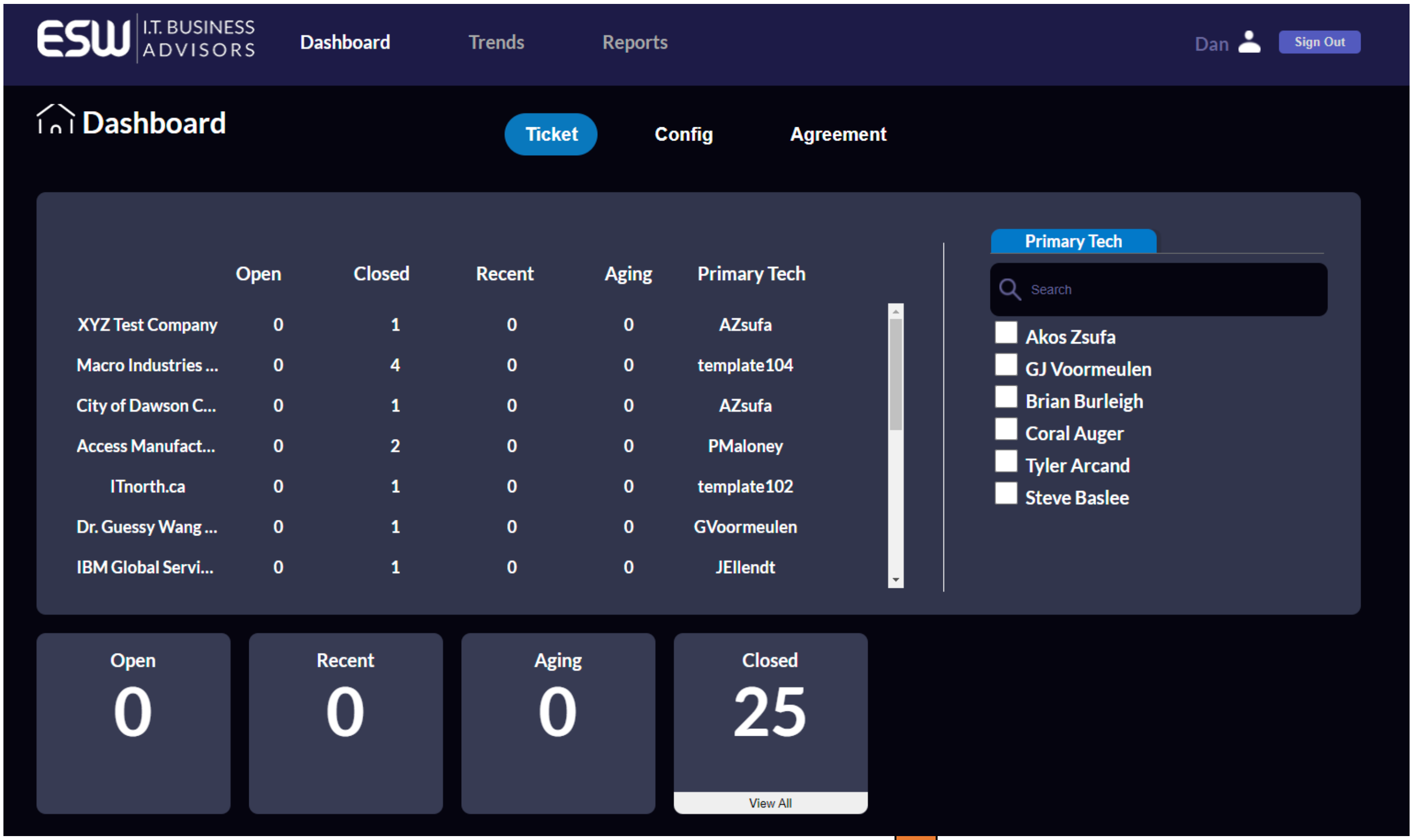
ESW I.T. BUISNESS ADVISORS

Reporting Portal User Guide

Mike Gartner
November 25, 2021

Main Dashboard



	Open	Closed	Recent	Aging	Primary Tech
XYZ Test Company	0	1	0	0	AZsufa
Macro Industries ...	0	4	0	0	template104
City of Dawson C...	0	1	0	0	AZsufa
Access Manufact...	0	2	0	0	PMaloney
ITnorth.ca	0	1	0	0	template102
Dr. Guessy Wang ...	0	1	0	0	GVoormeulen
IBM Global Servi...	0	1	0	0	JEllendt

□ Akos Zsufa
□ GJ Voormeulen
□ Brian Burleigh
□ Coral Auger
□ Tyler Arcand
□ Steve Baslee

Open 0

Recent 0

Aging 0

Closed 25

View All

1) Header and Navigation

Select “Dashboard”, “Trends”, or “Reports” to navigate to the corresponding interfaces. Clicking the “Sign Out” will sign out the current user, and return to the sign in screen.

2) Dashboard Navigation

Selecting “Ticket”, “Config”, or “Agreement” will sort the dashboard (3) by the corresponding data type. Selecting an option will also update (5) the ticket overview options to display the related details.

3) Dashboard

The scrollable dashboard can be filtered by using the dashboard filter (4) and a row may be selected to display additional details about that row.

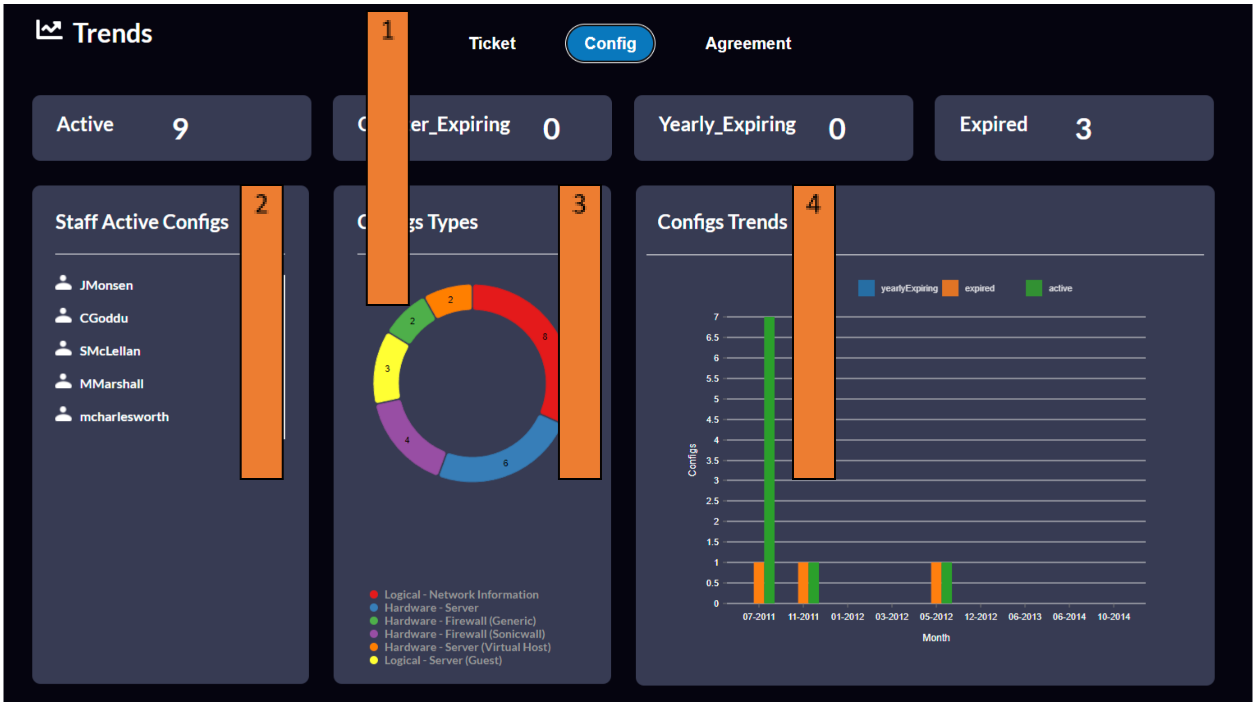
4) Dashboard Filter

The filter is used to filter the dashboard by the selected filter parameters. The search bar can be used narrow down the list of available filter options. Select a filter option by using the corresponding checkbox. Multiple filter options may be selected, displaying results for all selected filter options.

5) Overview

The overview section displays the totals of the selected data type organized by lifecycle state. Selecting “View All” will display a list of additional details for all of the related data.

Trends Dashboard



1) Trends Dashboard Navigation and Overview

Selecting “Ticket”, “Config”, or “Agreement” will sort the dashboard by the corresponding data type. Selecting an option will also update the rest of the dashboard with the related data. Below the navigation are totals of the selected data type organized by lifecycle state.

2) Staff List

This list displays all staff and the number of active cases of the selected type (see 1) that they are assigned to.

3) Pie Graph

The pie graph displays a breakdown of the selected data type (see 1) by the associated data subtypes. Hovering over a slice of the graph will display a tooltip of the name of the data subtype.

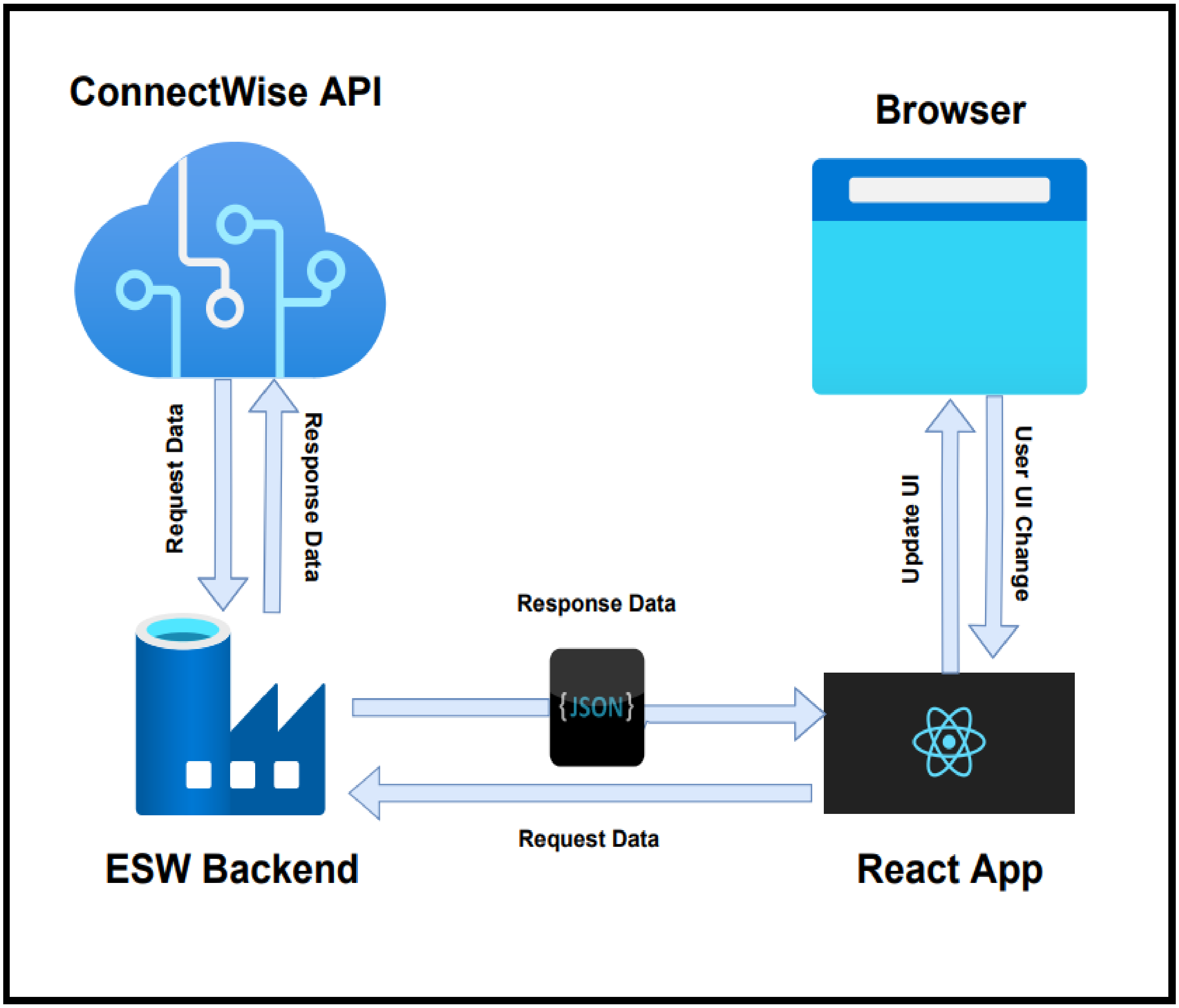
4) Bar Graph

The bar graph displays a month-by-month comparison of active and expired tickets. Hovering over a bar on the graph will display the associated month and totals of the selected data type for that month.

Transfer Documentation

The Big Picture

Below image shows the high-level flow and connections of the application and how it funnels information and commands to run:



Frontend Application

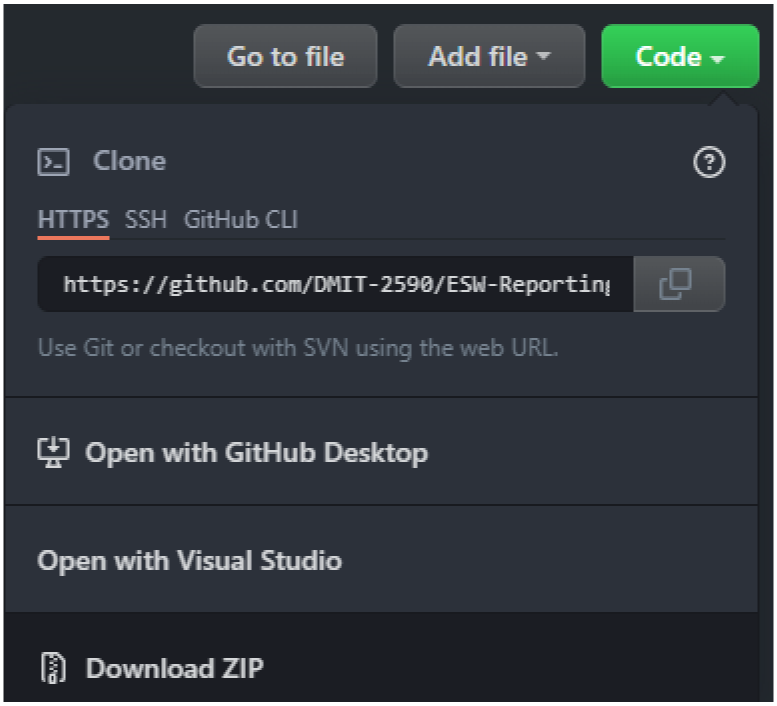
1) Source code

The source code for the frontend project can be found at the GitHub repository link: https://GitHub.com/DMIT-2590/ESW-Reporting-Portal. This repository is set as “Private” and access to the repository must be given to the users specifically for them to be able to access it.

2) Downloading (cloning) the application

To clone the application down to any local machine, you must either download the zip file directly from the GitHub repository’s page as shown in the below screenshot or have “Git” installed on your machine and run the following command:

```
git clone https://GitHub.com/DMIT-2590/ESW-Reporting-Portal.git
```



https://github.com/DMIT-2590/ESW-Reporting

Use Git or checkout with SVN using the web URL.

3) Downloading dependencies and running the app

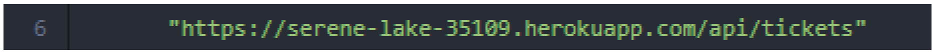
First, the machine running the app must have “node js” and “npm” installed before running it you can download them from their official pages by searching “download node js” and “download npm”. The following commands must run in the same order to first download all dependencies and then run the application (in the terminal or command line):

`npm install` (to download dependencies)
`npm start` (to run the app in localhost:3000)

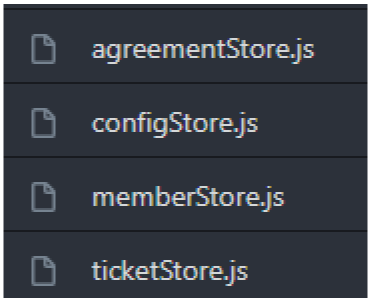
4) Linking the frontend app to the correct backend URL.

The app retrieves all relevant data from the backend application and changing where the backend application lives will require that the frontend app knows the new URL and is correctly configured to connect to it. The app has a folder in the file path of “ESW-Reporting-Portal/src/stores” which houses 4 store files as shown in the screenshot below, line 6 of each of these 4 files points to the URL of the backend application and must be updated anytime the backend application URL changes.

Please ensure that only the base URL is changed while keeping the endpoint the same. Example: In the below-shown URL. “https://serene-lake-35109.herokuapp.com” should be changed. But “/api/tickets” should NOT be changed.



```
"https://serene-lake-35109.herokuapp.com/api/tickets"
```



agreementStore.js

configStore.js

memberStore.js

ticketStore.js

Backend Application

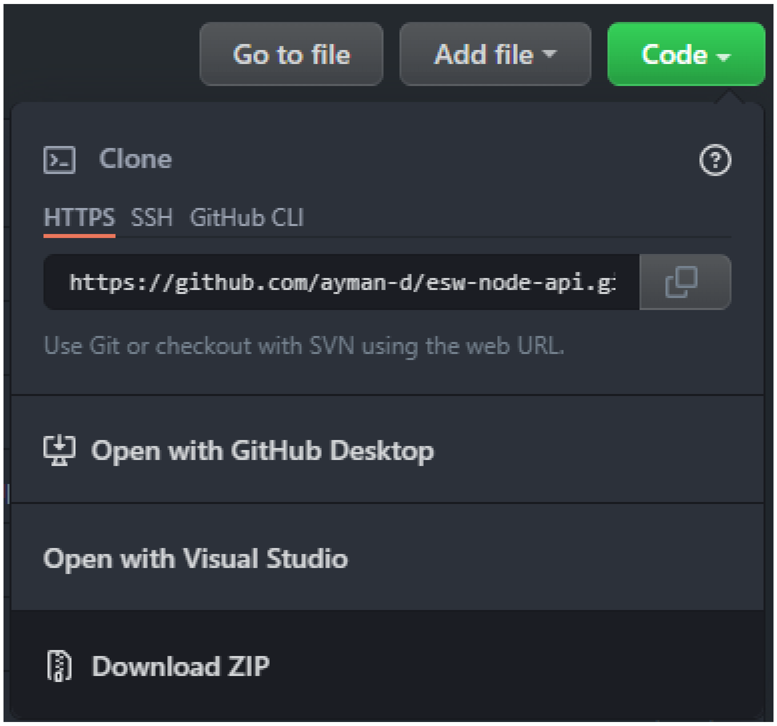
1) Source code

The source code for the frontend project can be found at the GitHub repository link: https://GitHub.com/ayman-d/esw-node-api. This repository is set as “Private” and access to the repository must be given to the users specifically for them to be able to access it.

2) Downloading (cloning) the application

To clone the application down to any local machine, you must either download the zip file directly from the GitHub repository’s page as shown in the below screenshot or have “Git” installed on your machine and run the following command:

```
git clone https://GitHub.com/ayman-d/esw-node-api.git
```



```
https://github.com/ayman-d/esw-node-api.g
```

Use Git or checkout with SVN using the web URL.

3) Downloading dependencies and running the app

First, the machine running the app must have "node js" and "npm" installed before running it you can download them from their official pages by searching "download node js" and "download npm". The following commands must run in the same order to first download all dependencies and then run the application (in the terminal or command line):

npm install (to download dependencies)
npm start (to run the app in localhost:5000)

4) Backend configuration file (config.env)

This file acts as the configuration that allows the backend application to talk to ConnectWise's API and retrieve the necessary data from it. Without correctly providing these 3 values in this file, the app will NOT be able to retrieve any data.

BASE_URL is the API URL provided by ConnectWise for data related to a specific customer, which in this case is ESW.

CLIENT_ID is provided by ConnectWise through their documentation. You can find it under their "CLIENT ID" page in the navigation of their REST API documentation page.

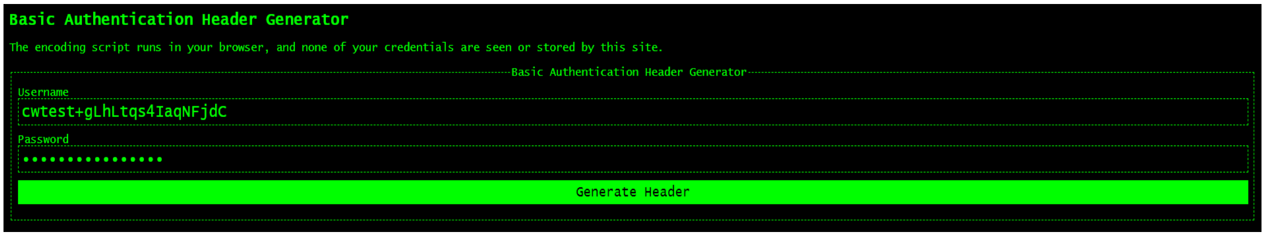
BASIC_TOKEN is the token generated by the keys they provide to us. The format that allows us to obtain the token is:

Username:cwtest+{{public key}}
Password:{{private key}}

The public and private key values should replace the values and without the double curly braces. Example (these values are fake):

Username:cwtest+gLhLtqs4IaqNFjdC
Password:flVRQAFDLQGDBB53

From those values, you can obtain the BASIC_TOKEN value by either using the Postman application and inputting them in the "Authorization" tab, or through online services such as https://www.blitter.se/utils/basic-authentication-header-generator/



Basic Authentication Header Generator

The encoding script runs in your browser, and none of your credentials are seen or stored by this site.

Basic Authentication Header Generator

Username
cwtest+gLhLtqs4IaqNFjdC

Password
••••••••••••••••

Once these 3 values of the config.env file have been updated, the rest of the backend application should continue running smoothly without having to change values in other files.

Security

1) Current Status

Currently, this app is in the proto-type stage and hence, no robust security has been implemented. The login functionality now works based on a file that exists in the backend application named "passwords.json" which simply has some values that the app will accept as authenticated users and their passwords.

2) Future Implementations

According to the client at the time of writing this document, ESW plans to implement security through their Azure Active Directory. This can be done by a few required steps:

Required by ESW:

a. On the Azure Portal, create an "app registration" on the Active Directory that has all the users that should have access to this app.

b. Once the app registration is created, you will gain access to a number of important values such as:

    I. Application (client) ID – represents the application talking to Azure (React)

    II. Directory (tenant) ID – represents the authority (Azure)

    III. Application ID URI – the same as client ID with api:// in front of it

c. The Application ID URI will not be created at first, and you must click on the option to "Add an Application ID URI to generate it.

d. Once that is done, you will have all of the required information, and you can provide the development team with the above listed 3 values

Required By Dev Team:

a. Follow all required steps in the documentation in this link: https://docs.microsoft.com/en-us/azure/active-directory/develop/tutorial-v2-react

b. The most important file in this documentation is "authConfig.js". To avoid confusion, here are the important values and where to input them for this to work

c. In the msalConfig object:

    I. clientId: Application ID URI

    II. authority: Directory (tenant) ID

    III. redirectUri: the URL of your frontend application (i.e. http://localhost:3000/ or the actual URL once it is hosted)

    IV. cacheLocation: "sessionStorage"

d. In the graphConfig object:

    I. graphMeEndpoint: "https://graph.microsoft.com/v1.0/me"