MSDF SDCOE Ed-Fi Data Hub Prototype:
Deployment Documentation

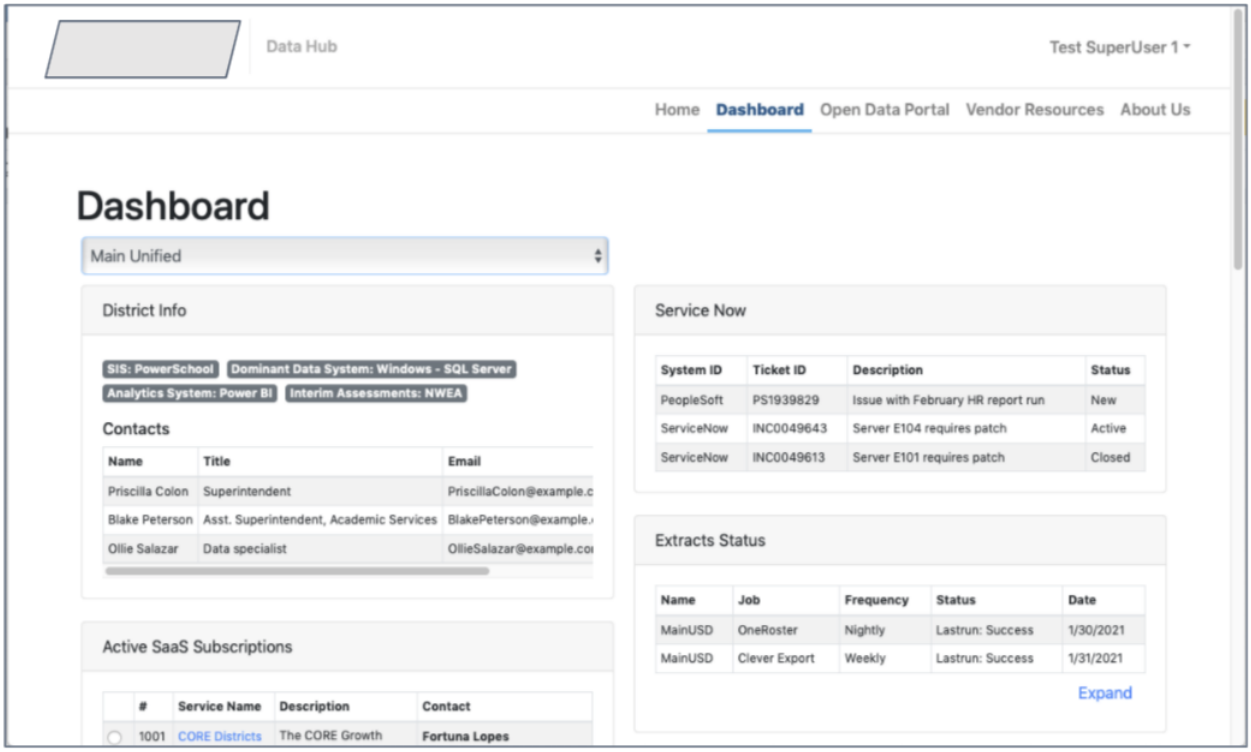
Revision July 2021
San Diego County Office of Education, Innovation and ITS (datascience@innovatesd.net)
Application development: Double Line, Inc.

Contents

Overview

The Data Hub prototype is a multifaceted yet easy-to-use interface between participating organizations (e.g. LEAs) and a collaborative host that manages Ed-Fi ODS's for members. The portal brings together automated Ed-Fi systems status with an ODS management request capability, access to registered applications, dashboards and services and participant contact information.

When a member organization logs in, the Data Hub displays any active alerts and provides a set of functional interactive "widgets": Ed-Fi status updates, Ed-Fi setup/maintenance requests, services & products selector and list of links to subscribed services (dashboards, 3rd party applications, etc.). The system has integrated email notifications for interactive features. Additional demonstration widgets include: a support-ticketing system, an ODS extract status list, and CRM-based contact list.



Key Features and Use Cases

Key Features

1) ODS monitoring features that leverage Ed-Fi API to obtain critical ODS information
2) Browser-based interface for external district users to view information about the various support relationships (persons, systems and processes) within the collaborative
3) System for handling district Ed-Fi requests
4) Data-driven back-end processes that represent API connections to district support systems
5) SSO using Azure AD for authentication and authorization for 3rd party applications

Use Cases

A collaborative member/district user would like to

- View active Ed-Fi ODS status, obtain API information
- Access a subscribed service (SaaS).
- Review active SDCOE support services for the district
- Review active support tickets or the status of extracts
- Subscribe/unsubscribe to an available service (SaaS) or SDCOE-offered support service.
- Request an ROP or Universal Transcript.

An collaborative host member user would like to:

- Review technical information collaborative members and personnel contact information (from CRM).
- View or update member/district subscribed SaaS products or collaborative services
- List member/district contacts

A Public user wants

- Review public data on the collaborative, area member districts and schools
- Access a public data endpoint
- Request an ROP transcript (past student)

Usage

The Data Hub prototype is a system that would be used both internally by staff within a collaborative host and externally by collaborative member / district leadership.

The initial idea was that a user could authenticate into the Data Hub, and that there would be the benefit of that SSO extending to analytical dashboards or even external applications where the SSO could extend the authentication for those systems -- a one stop place to get at a number of analytics, reports dashboards.

User Capabilities

A member/district viewer has access to their district information and cannot add/change any settings, but can view all and click on SaaS links, etc. This user is also able to access the separate transcript system and PowerBi dashboards run by the collaborative via Azure AD SSO.

A member/district Super User a staff person who has administrative access to their district's information and can make Ed-Fi requests and add/change services or SaaS subscription requests (adding a subscription does not activate it and make it live - it sends an 'order' or request' email to the collaborative host party responsible for the services or SaaS, and they set up any agreements/payment authorization and activate the service or system). This user is also able to access the separate transcript system and PowerBi dashboards run by the collaborative via Azure AD SSO.

An administrator superuser can access any district and view or make changes just like the district administrative user.

Prerequisites

For deployment, you will need a workstation or server running:

* Microsoft IIS web server

* Microsoft SQL Server instance

* Microsoft prerequisites

* An Azure AD subscription to support SSO (an alternative is to replace the SSO with local authentication)

For development, you will need:

* A local Microsoft SQL Server instance

* Visual Studio 2019 for the API backend

* Visual Studio Code for the React client web app

Deployment Instructions

Note: Also see README.md in GitHub repository for the project

1. Clone the git repo from the Ed-Fi Exchange: SDCOE Data Hub

2. Edit the following files:

    2.1. DataHub.Web/DataHub.Api/appsettings.json

        2.1.1. ConnectionStrings:Datahub - input the connection string for the prod database.
        "Server=[SERVER];Database=SDCOE_DataHub;User Id=sa;password=[PASSWORD]"

        2.1.2. Cors:AllowedOrigins - put the http(s) URLs for the site.
        "http://[URL]"

        2.1.3. Edit and Change the Following Settings

```
"SmtpSettings": {
   "EmailFrom": "[EMAIL FROM]",
   "EmailFromName": "[SITE NAME] Data Hub",
   "EmailTo": "[VALID EMAIL]",
   "EmailToName": "[SITE NAME] Administrator",
   "SmtpUser": "[SMTP USER NAME]",
   "SmtpPass": "[SMTP PASS]",
"EnableGlobalOverride": true,
   "GlobalEmailToOverride": "[VALID EMAIL]"
 },
 "OfferingsSettings": {
   "SendEmailOnParticipation": true
 },
```

    2.2. DataHub.Web/DataHub.Client.Web/src/Config.ts
        API_URL - the URL to the API
        export const API_URL = 'https://[SUBDOMAIN/SITE]**/api**'

    2.3. DataHub.Web/DataHub.Client.Web/src/utilities/AuthConfig.ts
        auth:redirectUri - the redirectURI for the app to send the user to after sign in
        "https://[SUBDOMAIN/SITE]/"

3. In terminal change to the DataHub\DataHub.Web directory and restore and build the project

   cd .\DataHub.Web
   dotnet restore DataHubApi.sln
   dotnet build DataHubApi.sln --configuration release

4. Zip the api build at the following directory:

   DataHub\DataHub.Web\DataHub.Api\bin\Release\netcoreapp3.1\
   to
   DataHub\DataHub.Web\DataHub.Api\bin\Release**netcoreapp3.1.zip**

5. Change to the DataHub\DataHub.Web\DataHub.Client.Web directory and install and build yarn

   cd DataHub.Client.Web
   yarn install
   yarn build

6. Zip the website build at the following directory:

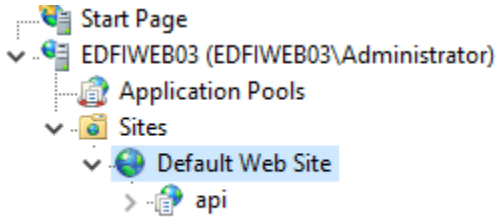
   DataHub\DataHub.Web\DataHub.Client.Web\build
   to
   DataHub\DataHub.Web\DataHub.Client.Web**build.zip**

7. Copy **netcoreapp3.1.zip** and **build.zip** to the web server and extract them to

   build.zip → C:\Datahub*
   netcoreapp3.1.zip → C:\Datahub\api*
   ***** if deploying onto existing site, IIS may need to be stopped before copying

8. Deployed Solution in IIS

   8.1. Deployment in IIS Manager shown below



Customizing Deployment for Local Use

Please review the README.md file in the repository for the latest version updates. Below are customizations that can be set up within the application.

Application Name, Logos and Coloration

- Find the application name and logo image file references in `/DataHub.Web/DataHub.Client.Web/public/manifest.json` and look for `[YOUR ORGANIZATION]` to replace in the JSON. There are references to locations in the folder for `favicon.ico`, a 192x192 `logo192.png`, and a 512x512 `logo512.png`. Notes:
    - The main logo referenced in the application code is the file “Logo.png” within the project.
    - Two other logo files within the project are “header-logo.svg” and footer-logo.svg”.
    - The favicon is the file “favicon.ico” within the project.
- Color settings for the UI are contained in `/DataHub.Web/DataHub.Client.Web/index.css`
- To update the interface to include your organization name and contact info, these can be updated within `DataHub.Web/DataHub.Client.Web/src/components/Footer.tsx` where you can replace [YOUR ORGANIZATION], [ADDRESS], [CITY-STATE-ZIP] and [PHONE].
- There may be locations in code and content where `SDCOE` is included for display purposes. These may be replaced (when not part of application code) with another collaborative or organization’s name or abbreviation.

Application URL

- Update application URL within `/DataHub.Web/DataHub.Client.Web/src/components/Navigation.tsx` so that the application refers to the domain and base URL for your installation. Look for [YOUR SITE URL].

Sample database contents:

- The file `/DataHub.Migrations/sql/runFirstAfterUp/0011_CreateDemoData.env.LOCAL.sql` populates many tables with sample data. It is a good place to start if there is the

desire to customize an installation. There are a number of table INSERT statements here that can be altered/updated to suit your needs.
    - Look for `[ACCOUNT]`, `[YOUR ORGANIZATION WEB SITE]` and `[COLLABORATIVE DOMAIN]` for easy-to-find replacements.
    - Note that the names, email addresses and phone numbers have been randomly generated.

User emails, authentication

- Azure AD settings: see `OpenId` block in `/DataHub.Web/DataHub.Api/appsettings.json`
- SMTP user settings: see `SmtpSettings` block in `/DataHub.Web/DataHub.Api/appsettings.json`

SMTP Server

- If you would like to change the default SMTP service (MailJet), this can be updated in: `/DataHub.Web/DataHub.Api/Services/EmailService.cs`