Date :  04-Sep-23

GITHUB Data Adapter

version:1.0.1

# **Overview**

A GitHub Data Adapter is a RAML based adapter that supports more than 200 APIs.

The following sections help you to use the GitHub Data Adapter component in your Volt Foundry application.

# **Getting Started**

## **Prerequisites**

Before you start using the GITHUB data adapter, ensure you have the following:

* [HCL Foundry](https://manage.hclvoltmx.com/)
* Volt MX Iris
* A [GitHub Inc.](https://github.com/) account.
* An app registered in the GitHub account.
* Client ID and Access key as provided by GitHub Application.

## **Platforms Supported**

### Mobile

#### iOS

#### Android

## **Importing the adapter**

**To import the Data Adapter to Volt Foundry, do the following:**

1. Sign in to the  [HCL Foundry](https://manage.hclvoltmx.com/).
2. From the left navigation menu, select **API Management**.
3. In **API Management**, select **Custom Data Adapters**.  
   
4. Click **IMPORT** to import a custom data adapter.  
   
5. On the Import Data Adapter dialog box, click browser to import.  
   
6. Selected GITHUB data adapter zip file and click **IMPORT**.

After you import the data adapter, Volt Foundry opens a window that shows the metadata of the data adapter.

A screenshot of a web page

Description automatically generated

After you import the data adapter, you can view it on the Custom Data Adapters page and use it to create services on Volt Foundry.

## **Creating an Identity Service with GitHub**

GitHub provides support for OAuth2.0. We need to configure the identity in order to access the GitHub data with less rate limit.

Create an Identity service with service type as **OAuth2.0** to connect with your GitHub site and respective account.

**To create an Identity service with GitHub**

1. Log on to your Volt Foundry. The **Dashboard** page appears by default.
2. In the left pane, click the **API Management** menu. The **APIs** tab opens by default.
3. Click the **Identity** tab. The **Identity** tab opens with a list of existing identity services.
4. Click **CONFIGURE NEW**. The **Identity Service** tab opens.
5. In the **Name** box, type a unique name for your service.
6. From the **Type of Identity** list, select **OAuth2.0**.
7. In the **Client ID** box, type the Client ID provided by GitHub
8. Type the URL in the **Authorize Endpoint** box and **Token Endpoint** box. These fields are mandatory. In the **Client Secret** box, type the secret key provided by GitHub. Type the

A screenshot of a computer

Description automatically generated

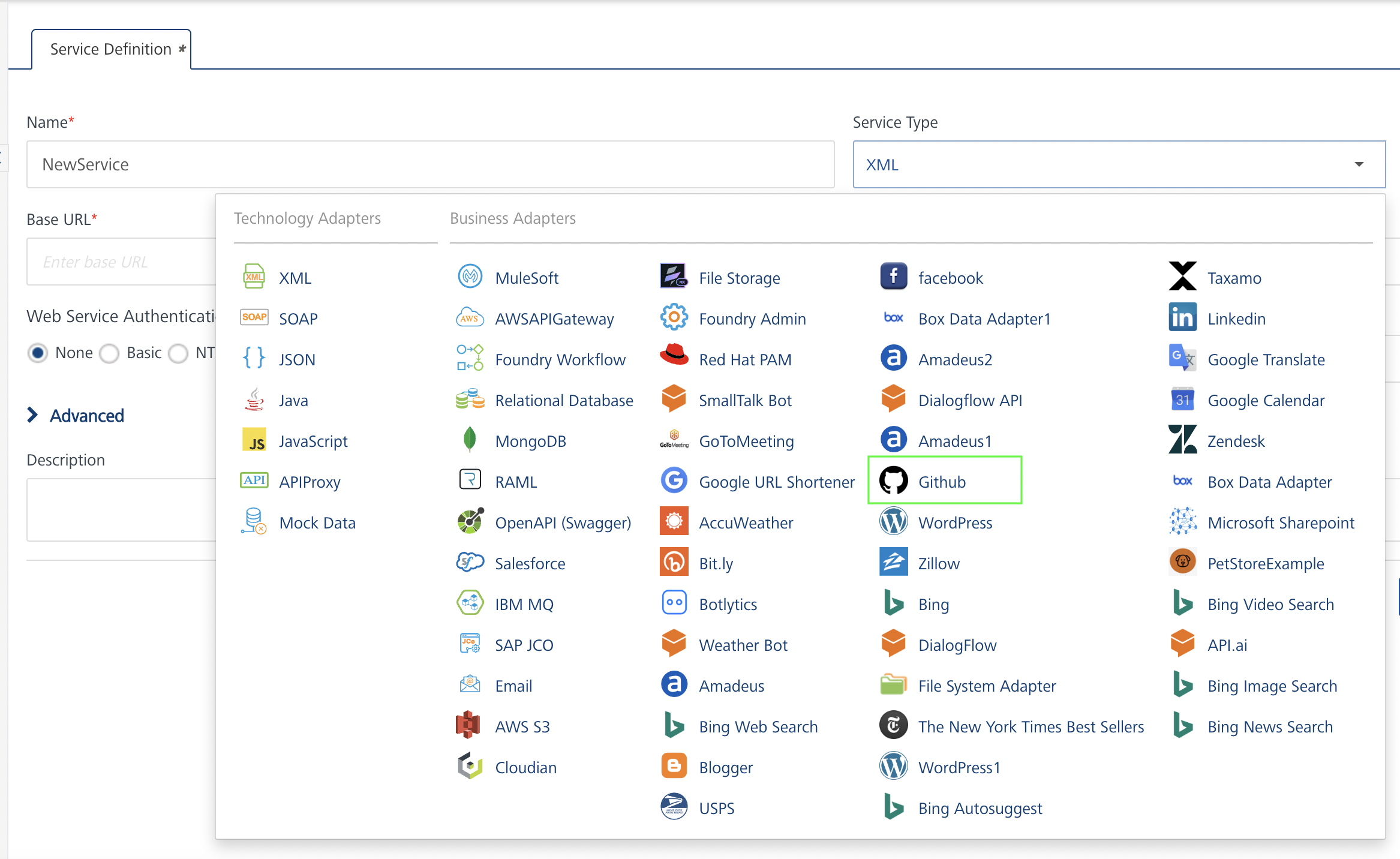
1. Click **Save**.

If the details provided are valid, the Volt Foundry application authenticates to your GitHub Repository and allows you to make the API calls.

## **Creating an Integration Service with Github**

After you import GitHub Data Adapter into Volt Foundry, you must create an integration service with service type as **GitHub** to connect with your GitHub Repository.

**To create an integration service with GitHub, do the following:**

1. Log on to your Volt Foundry. The **Dashboard** page appears by default.
2. In the left pane, click the **API Management** menu. The **APIs** tab opens by default.
3. Click the **Integration** tab. The **Integration** tab opens with a list of existing integration services.
4. Click **CONFIGURE NEW**. The **Service Definition** tab opens.
5. 
6. To create a new service, click the **+** button or the **CONFIGURE NEW** button.  
   
7. On the Service Definition tab, select the service type as GITHUB, and click **SAVE**.  
   

A screenshot of a computer

Description automatically generated

1. In the **Name** box, type a unique name for your service.
2. From the **Service Type** list, select **GitHub**.

**Note:**The **GitHub** is listed only after you import the GitHub Data Adapter into Volt Foundry

1. Select **Use Existing Identity Provider**.
2. Select **github** from the drop down list and click **Test Login**.
3. On successful test log on, click **Save**.

If the details provided are valid, the Quantum Fabric service connects to your GitHub repository and allows you to make the API calls.

Refer to [Integration Services](http://docs.kony.com/konylibrary/mobilefabric/kony_mobilefabric_user_guide/Default.htm#Services.htm) for more information on creating and using integration services.

## **Creating an Operation**

In the Volt Foundry console, to make any API call, you must create an operation for the respective API and then execute the operation. Executing an operation involves making the API call and displaying the response. For more information, see [Executing an Operation](https://docs.kony.com/marketplace/V8Marketplace/Content/Marketplace/github.htm#Performi).

**Procedure for User to log on to his GitHub account and view his/ her repositories:**

1. Log on to your Volt Foundry. The **Dashboard** page appears by default.
2. In the left pane, click the **API Management** menu. The **APIs** tab opens by default.
3. Click the **Integration** tab. The **Integration** page appears with a list of existing integration services.
4. From the list of integration services, select the integration service that you have created with service type, **GitHub**. The **Service Definition** tab of the selected integration service opens by default.
5. Click the **Operations List** tab. The **Operations List** tab opens.
6. Click the context menu arrow to expand the **Please Select** list. The **GitHub API** will be displayed.
7. Click the Plus icon to expand **GitHub API**.
8. Click the Plus icon to expand **Services**, and select the required operation.

A screenshot of a computer

Description automatically generated

You can create multiple operations with required APIs and save them under **Configured Operations** section. You can then execute the operations when ever required.

## **Executing an Operation**

Executing an operation involves making the API call by setting the necessary parameters and displaying the response. The procedure to execute any operation is same, but the request parameters of the APIs vary. For more information on request parameters of each API, refer [GitHub API Catalog](https://developer.github.com/v3/repos/). This section provides you the procedure to execute the search of a GitHub repository which is explained in [Creating an Operation](https://docs.kony.com/marketplace/V8Marketplace/Content/Marketplace/github.htm#Creating).

**Procedure to execute the get Repository Search API:**

1. Log on to your Volt Foundry. The **Dashboard** page appears by default.
2. In the left pane, click the **API Management** menu. The **APIs** tab opens by default.
3. Click the **Integration** tab. The **Integration** tab opens with a list of existing integration services.
4. From the list of integration services, select the integration service that you have created with service type, **GitHub**. The **Service Definition** tab of the selected integration service opens by default.
5. Click the **Operations List** tab.
6. Under the **Configured Operations** section, click the **get/search/repository** operation. A separate tab opens along with Request Input sub-tab opened.
7. In the **Name** box, the name of the operation is displayed by default. If you want, you can change the name.
8. On the **Request Input** tab, in the Body section, for the following parameters, specify the values in the respective boxes under the **Default Value** column.
   * **q:** specify the name of the repository location.
   * **sort:** specify the type of sorting (fork).
   * **order:** Specify the default value as ascending (asc) or descending order.

A screenshot of a computer

Description automatically generated

Click **SAVE AND FETCH RESPONSE**. The **Output Result** dialog appears with the data requested for. Otherwise, the **Output Result** shows an error.

A screenshot of a computer program

Description automatically generated

## **Publishing the App in Foundry**

After adding the **Github** component to your app and configuring the necessary configurations, you must publish the app to Volt Foundry. For more information, refer to Publish a Project to Volt Foundry.

# **References**

## **Events**

-- None of the events are exposed.

## **API’s**

-- None of the APIs are exposed.

# **Revision History**

App version 1.0.0

## **Known Issues**

None.