Lab 04: Enhancing Data with Cloud Sources

Technologies showcased: ADF GUI, Azure Storage, HTTP Source. Copy Data Wizard

Author: Joshuha Owen

Date: 12/13/2017

Table of Contents

[Summary 3](#_Toc502319916)

[Pre-requisites 4](#_Toc502319917)

[Scenario 4](#_Toc502319918)

[Part 1 – Creating the Pipeline to Get REST Data 4](#_Toc502319919)

[Part 2 – Calling a Execute Pipeline from another Pipeline 12](#_Toc502319920)

## Summary

This tutorial walks through creating a pipeline copy activity to copy web restful API data to a local file in Azure blob storage to join with transforms in a later step.

In this lab we will:

* Show the Copy Data wizard to configure the pipeline
* Configure the HTTP Source
* Chain one pipeline to another using the Execute Pipeline activity

In this lab we will be using the following REST API information to get weather data. Note that this data was randomly generated.

REST APU URL:

<https://weather-b362.restdb.io/rest/random-weather-dec-2016>

Header Information:

x-apikey: 5a31643dcb25c1076c7cd9ab

Content-Type: application/json

## Pre-requisites

* Azure Subscription with rights to use/deploy Azure services, and X of Azure credit
* Azure Blob storage container
* Restful API configured for GET access with key
* Azure Data Factory

## Scenario

|  |  |  |
| --- | --- | --- |
| Part 1 – Creating the Pipeline to Get REST Data | | |
| **Scenario** | | |
| We will be setting up a new pipeline to copy weather data from our REST API into our Azure blob storage account. | | |
| **Commentary / Notes** | **Click Steps & ‘Bits’** | **Screenshots** |
| We will be using the Azure Data Factory we created in Lab 01 called adflab-adf.. | 1. Navigate to the Azure portal within your web browser and navigate to <https://portal.azure.com> 2. Open the Azure Data Factory blade [adflab-adf], pinned from a previous lab but if not navigate to it using the All Resources menu item. | *TO-DO: Replace with GUI Pipeline editor tile exists* |
| !!! GUI tile button doesn’t exist yet!!! | 1. In the Overview blade you should see the following Quick Links: | TO-DO: Replace when GUI Pipeline tile exists |
| !!! GUI tile button doesn’t exist yet!!! | 1. Click the Pipeline Editor button and you should see graphical user interface Overview page. |  |
| We’ll use the Copy Data wizard to walk us through configuring this pipeline. | 1. Click the Copy Data button under Let’s Get Started. |  |
|  | 1. You should now see Copy Data wizard. 2. Fill out the Task Name. We will be using Weather to Blob Copy 3. Click Next. |  |
|  | 1. For the Source, click on the HTTP source. 2. Click Next. |  |
|  | Fill out the following information: Connection Name: HTTP-Weather URL: <https://weather-b362.restdb.io/rest/random-weather-dec-2016>   1. Click the Advance Settings section to expand. |  |
|  | 1. Verify the HTTP Method selection show Get.   For the additional headers input put in the following: x-apikey: 5a31643dcb25c1076c7cd9ab  Content-Type: application/json   1. Click Next. |  |
| As this REST API contains an array we are using array of object. See for a list of different file formats from the HTTP connector. | 1. For File Pattern, switch the input to Array of Objects. 2. This should cause an auto-refresh and you should see the data appear under Preview and Schema. We will keep the default columns identified. 3. Click Next. |  |
| Since we already created the Connection for the dataset in Lab 3 we will be using that connection here. | 1. Click the From Existing Connections tab. 2. Click AzStorage-Staging connection. 3. Click Next. |  |
|  | 1. Verify the Azure Storage Service shows Azure Blob. 2. Click Next. |  |
|  | 1. Click Browse to browse our Storage location. 2. Choose Input for the folder. 3. For the Filename put in 201612-WeatherData.txt 4. Click Next. |  |
|  | 1. Check the box that says Add header to file. 2. Click Next. |  |
| Here you could change the Actions to ignore bad rows or log bad rows. | 1. We won’t change any settings here. 2. Click Next. |  |
|  | 1. You should now see a Summary page of the Pipeline and Connections being created. 2. Click Next. |  |
| After this is run you can use Azure Storage Explorer to verify you have a 20612-WeatherData.txt in your Input container. | 1. Your pipeline show now be deploying. 2. After deploying this will also run the pipeline. 3. Close this window or click on Click here to visualize copy pipeline to return to Pipeline GUI. |  |

|  |  |  |
| --- | --- | --- |
| Part 2 – Calling a Execute Pipeline from another Pipeline | | |
| **Scenario** | | |
| We will be using the pipeline we created in Lab 03 and call an execute pipeline to call the pipeline we just created. This will then create a chain of pipeline. | | |
| **Commentary / Notes** | **Click Steps & ‘Bits’** | **Screenshots** |
| This concept is similar to an Execute Package task in SSIS if you are familiar and would allow the creation of complex controller pipeline logic if you need that. | 1. You should now be back at the Pipeline editor screen. If not click on the Pencil icon in the left menu and click the S3 to Blob Copy Pipeline we created in Lab 03. |  |
|  | 1. In the Pipeline GUI, drag the Execute Pipeline activity to the right of the Send Success Email activity. 2. Rename the activity to Exec Weather to Blob Copy. 3. Click Save. |  |
|  | 1. Drag from the green square of the Send Success Email activity to the Execute Pipeline activity. 2. Click the Exec Weather to Blob Copy activity. |  |
| After this step we now have a chained pipeline that after copying the S3 data to our staging area also copies weather data to our staging area. | 1. Click the Settings tab. 2. In the Invoked Pipeline dropdown select the Weather to Blob Copy pipeline we created in Part 1 of this lab. 3. Click Save. |  |

**IMPORTANT: AVOID INCURRING EXTRA CHARGES BY PAUSING YOUR SUBSCRIPTION RESOURCES**