# Using TOM to Create a DirectLake Dataset

This repository contains a Fabric notebook with Python code and a C# console application which can be used to create a DirectLake data model using the Tabular Object Model (TOM).

Steps to completing this demonstration:

* Create workspace associated with Fabric capacity
* Create a new Lakehouse in the new workspace
* Create Lakehouse tables using a pre-provided Fabric notebook
* Run custom application to create DirectLake data model using TOM

Now we will go through steps.

## Create workspace associated with Fabric capacity

* Get URL to Workspace Connection
* Write down workspace name

## Create a new Lakehouse in the new workspace

* Get Lakehouse SQL Endpoint

## Create Lakehouse tables using a pre-provided Fabric notebook

* Upload Python notebook named **CreateLakehouseTables.ipynb**
* Associate notebook named **CreateLakehouseTables.ipynb** with Lakehouse
* Execute code in notebook to copy CSV files from GitHib repository into Lakehouse file system
* Execute code in notebook to load CSV files and convert then into delta tables for bronze zone
* Execute code in notebook to load bronze tables and reshape/transform data in into delta tables for silver layer
* Execute code in notebook to generate calendar table for silver layer

## Run custom application to create DirectLake data model using TOM

* Create Azure AD application
  1. Create a native/public application with redirect URI of <http://localhost>
  2. Record Application ID for use in console application.
* Download C# console application source code and open project in Visual Studio 2022
* Open **AppSettings.cs** and updae the following:
  1. ApplicationID of Azure AD application
  2. Workspace Connection
* SQL Endpoint
  1. Lakehouse Name
  2. UserID and Password to prevent interactive login
  3. Save changes
* Run application
  1. It should run without error
  2. When done, verify you can see new data model and use it to create new report