Azure Meal Project | ETL Documentation

## Azure Meal Project | ETL Documentation

#### Overview

The Meal Project is a cloud-based ETL solution hosted on Azure. It efficiently manages raw data stored in a blob container, processes the data using Function Apps, and ingests the results into a SQL Server database. This processed data is then connected to Power BI, where a report is generated to provide valuable insights. Figure 1 illustrates the architecture diagram of the pipeline.



Figure 1: Architecture Design of Meal Project ETL

#### Blob Storage

The raw data from the user or client is stored in Azure Blob Storage, an object storage solution that accommodates various file types, such as .csv and .txt files, organized into separate folders. Given that our input files are primarily .csv, this service provides an optimal solution for our needs. Additionally, Azure Blob Storage is easily accessible through the Microsoft Azure Storage Explorer, which offers a user-friendly interface for managing data. Users can utilize this tool to upload and review files within their designated folders, as illustrated in Figure 2.

In our project, we have created two directories into the blob container ‘csv-files’ named

1. Archived: contains files that are processed
2. InputFiles: contains file that will be processed and be sent to achieve



Figure 2: Microsoft Storage Explorer view of blob-container ‘csv-files’

#### FunctionApp

The Azure Function App is a serverless service that executes in response to specific events. Our Function App is configured with a timer trigger that activates every 15 minutes. The entire setup has been developed using Visual Studio Code, where we manage the repository and handle deployments. You can find the repository at the link below.

Repo Link: <https://drive.google.com/file/d/1eCUw2L2qf7bSXF1fZcBo62LdRNo66Sad/view>

 The code runs in the following sequence

1. Look for the input file in the ‘InputFiles’ directory in blob container
2. Process the data based on ‘WLD’ columns and the defined calculations
3. Ingest the data into Azure SQL Server Database by making connection
4. Move the files from ‘InputFiles’ to ‘Archived’ directory after successful data ingestion

####

#### Azure SQL Server Database

Our final processed data is stored in Azure SQL Server Database. Here we have declared two tables

1. dbo.my\_table: The main table where our final processed data is in place and appended after every Function App execution
2. dbo.my\_table: It is created for testing process to validate the data ingestion coming from functionApp so that the data integrity of main data is maintained

The schema for both tables are same and includes the following columns

1. variety
2. formulation
3. no\_of\_samples
4. wLD0 to wLD99
5. first\_day
6. last\_day
7. total\_days
8. avg\_weight\_loss\_rate
9. avg\_weight\_loss

#### PowerBI Report

In PowerBI, we have linked the data from the SQL Server Database and transformed it using different DAX queries. As a result, we summarized the data as per the need of the desired visual and created different charts as a draft report shown in figure 3



Figure 3: PowerBI draft report for our final data from Azure SQL Server Databases