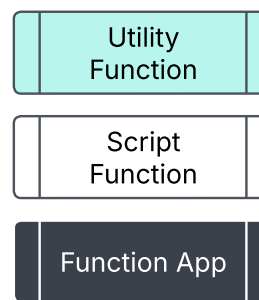


Prod Datalake Project

Azure Function App



Tech Stack



Function Apps

- Title:** elc_prod_items
Author: Dominic Vicchiollo
Dates: 2025-03-17
Trigger Schedule: Mon-Fri hours 6am -5pm every hour
Description: Pulling data from Mongo items collection, parsing and sending parquet file to data lake
- Title:** elc_prod_postings
Author: Dominic Vicchiollo
Dates: 2025-03-17
Trigger Schedule: Mon-Fri hours 6am -5pm every hour
Description: Pulling data from Mongo postings collection, parsing and removing html formatting and sending parquet file to data lake
- Title:** elc_prod_postings_note
Author: Dominic Vicchiollo
Dates: 2025-03-18
Trigger Schedule: Mon-Fri hours 6am -5pm every hour
Description: Pulling data from Mongo postings collection, filtering on the notes column and parsing array, as well as removing html formatting
- Title:** elc_prod_orders **multiple**
Author: Dominic Vicchiollo
Date: 2025-05-28
Trigger Schedule: Mon-Fri hours 6am each day
Description: Pulling data from Mongo orders collection, historical ran/updated in datalake, 6 separate instances scheduled 5 min apart for last 6 months (including this month) to catch any changes/updates to previous records/shippments

Package Directory

- function_app.py** - function app file hosting timer triggers for utils functions
elc_prod_postings.py
elc_prod_items.py
elc_prod_postings_note.py
- utilities** - Function modules used within package
datalake_import.py - SQL function to manage the import of parquet files to the data lake
elc_prod_items.py - Function to call elc-prod items collection, define headers and data types and perform transformations
elc_prod_postings.py - Function to call elc-prod postings collection, define headers and data types and perform transformations
elc_prod_postings_note.py - Function to parse postings - note column and normalize to its own parquet file in data lake
synapse_sql_connector.py - SQL function to manage table creation, column selection and data types
elc_prod_orders.py - Script to run filtered query on mongoDB to catch all orders for last 6 months and update order files in datalake
__init__.py

azure function apps v2 using implicit decorators requires binding file to be named as function app, and underlying functions to be identified in decorators

function_app.py

Orders Collection - running last six months on separate instances, historical ran once already, can be ran locally to update if needed.

