fabric_remote_tools: OneLake CRUD Operations

Renan Peres

2024-07-01

Install Package

```
%pip install https://github.com/renan-peres/fabric-remote-tools/raw/main/fabric_remote_tools-
```

Import Modules & Authenticate

```
from fabric_remote_tools import FabricAuth, OneLakeUtils
import os
from dotenv import load_dotenv
load_dotenv()

# Load Fabric Environmet Variables (.env File)
account_name = os.getenv("ACCOUNT_NAME")
workspace_id = os.getenv("WORKSPACE_ID")
lakehouse_id = os.getenv("LAKEHOUSE_ID")

# Get Authentication Token
token = FabricAuth.get_service_principal_token()

# Get File System Client
file_system_client = FabricAuth.get_file_system_client(token, account_name, workspace_id)
```

Write to Lakehouse (Files/Tables)

Local Tables (Delta)

```
# Single Table
OneLakeUtils.write_to_lakehouse(
    file_system_client=file_system_client,
    lakehouse_id=lakehouse_id,
    upload_from="local",
    source_path="../assets/data/Tables/venture_funding_deals_delta",
    target_path="Tables/local_venture_funding_deals_delta"
)

# Multiple Tables in a Folder
OneLakeUtils.write_to_lakehouse(
    file_system_client=file_system_client,
    lakehouse_id=lakehouse_id,
    upload_from="local",
    source_path="../assets/data/Tables",
    target_path="Tables/",
)
```

Local Files/Folders

```
# Whole Folder
OneLakeUtils.write_to_lakehouse(
    file_system_client=file_system_client,
    lakehouse_id=lakehouse_id,
    upload_from="local",
    source_path="../assets/data/Files",
    target_path="Files/"
)

# Individual Subfolder inside a Folder
OneLakeUtils.write_to_lakehouse(
    file_system_client=file_system_client,
    lakehouse_id=lakehouse_id,
    upload_from="local",
    source_path="../assets/data/Files/Contoso",
    target_path="Files/Contoso"
```

```
)

# Specific File in a Folder
OneLakeUtils.write_to_lakehouse(
    file_system_client=file_system_client,
    lakehouse_id=lakehouse_id,
    upload_from="local",
    source_path="../assets/data/Files/Contoso/contoso_sales.csv",
    target_path="Files/Contoso/contoso_sales.csv",
)
```

GitHub (Public Repo)

```
# Whole GitHub repository
OneLakeUtils.write_to_lakehouse(
    file_system_client=file_system_client,
    lakehouse_id=lakehouse_id,
    upload_from="github",
    source_path="https://github.com/renan-peres/Polars-Cookbook.git",
    target_path="Files/GitHub/Polars-Cookbook"
)

# Single Table (Delta) in Repository
OneLakeUtils.write_to_lakehouse(
    file_system_client=file_system_client,
    lakehouse_id=lakehouse_id,
    upload_from="github",
    source_path="https://github.com/renan-peres/Polars-Cookbook.git",
    target_path="Tables/github_venture_funding_deals_delta",
    folder_path="data/venture_funding_deals_delta"
)

# Specific folder from GitHub repository
OneLakeUtils.write_to_lakehouse(
    file_system_client=file_system_client,
    lakehouse_id=lakehouse_id,
    upload_from="github",
    source_path="https://github.com/renan-peres/Polars-Cookbook.git",
    target_path="Files/GitHub/data",
```

```
    folder_path="data"
)
```

GitHub (Private Repo)

```
github_token = os.getenv("GITHUB_PERSONAL_ACCESS_TOKEN")
github_username = os.getenv("GITHUB_USERNAME")
gh_repo_name = os.getenv("GITHUB_REPO_NAME")

# Whole GitHub private repository
OneLakeUtils.write_to_lakehouse(
    file_system_client=file_system_client,
    lakehouse_id=lakehouse_id,
    upload_from="github_private",
    github_token=github_token,
    github_username=github_username,
    repo_name=gh_repo_name,
    target_path=f"Files/GitHub/{gh_repo_name}"
)

# Specific folder from GitHub private repository
OneLakeUtils.write_to_lakehouse(
    file_system_client=file_system_client,
    lakehouse_id=lakehouse_id,
    upload_from="github_private",
    github_token=github_token,
    github_username=github_username,
    repo_name=gh_repo_name,
    target_path="Files/GitHub/data",
    folder_path="data"
)
```

Azure DevOps (Private Repo)

```
organization_url = os.getenv("ORGANIZATIONAL_URL")
personal_access_token = os.getenv("PERSONAL_ACCESS_TOKEN")
project_name = os.getenv("PROJECT_NAME")
repo_name = os.getenv("REPO_NAME")
```

```
# Whole Azure DevOps repository
OneLakeUtils.write_to_lakehouse(
    file_system_client=file_system_client,
    lakehouse_id=lakehouse_id,
    upload_from="azure_devops",
    project_name=project_name,
    repo_name=repo_name,
    organization_url=organization_url,
    personal_access_token=personal_access_token,
    target_path=f"Files/AzureDevOps/{repo_name}",
)

# Specific folder from Azure DevOps repository
OneLakeUtils.write_to_lakehouse(
    file_system_client=file_system_client,
    lakehouse_id=lakehouse_id,
    upload_from="azure_devops",
    project_name=project_name,
    repo_name=repo_name,
    organization_url=organization_url,
    personal_access_token=personal_access_token,
    target_path="Files/AzureDevOps/data",
    folder_path="/data",
)
```

List Items from Lakehouse (Files/Tables)

```
# List All Items in Lakehouse
OneLakeUtils.list_items(
    file_system_client=file_system_client
    ,lakehouse_id=lakehouse_id
    ,target_directory_path="Tables" # Tables or Files
    #  ,print_output= True # Optional
)
```

Delta Table Operations

Read Delta Table from Lakehouse

```
from fabric_remote_tools import FabricAuth, OneLakeUtils
import os
from dotenv import load_dotenv
load_dotenv() # Load environment variables from .env file

# Authenticate and obtain access token
file_system_client = FabricAuth().get_client_secret_token()

# Read Table from Lakehouse into Dataframe
workspace_name = os.getenv("WORKSPACE_NAME")
lakehouse_name = os.getenv("LAKEHOUSE_NAME")
table_name = "Tables/venture_funding_deals_delta_partitioned"
table_path = f"abfss://{workspace_name}@onelake.dfs.fabric.microsoft.com/{lakehouse_name}.La

df = OneLakeUtils().read_delta_from_fabric_lakehouse(
    file_system_client=file_system_client,
    table_path=table_path,
    engine='duckdb',  # Supported options: 'duckdb', 'polars'
    version=11,  # Optional: specify the version to read
    # row_limit=10  # Optional
)

display(df)
```

```
FloatProgress(value=0.0, layout=Layout(width='auto'), style=ProgressStyle(bar_color='black'))
```

Company varchar	Amount varchar	Lead investors varchar	…	Date reported varchar	Day int8	Month int8
Rapport Therapeutics	$100,000,000	Third Rock Venture…	…	3/7/23	7	3
Character.AI	$150,000,000	Andreessen Horowitz	…	3/21/23	21	3
Palmetto	$150,000,000	TPG Rise Climate	…	3/6/23	6	3
Consensus	$110,000,000	Sumeru Equity Part…	…	3/8/23	8	3
Bicara Therapeutics	$108,000,000	Red Tree Venture C…	…	3/6/23	6	3
CARGO Therapeutics	$200,000,000	Third Rock Venture…	…	3/1/23	1	3
Humane	$100,000,000	Kindred Ventures	…	3/8/23	8	3

Rippling	$500,000,000	Greenoaks	…	3/17/23	17	3
Amogy	$139,000,000	SK Innovation	…	3/22/23	22	3
Adept AI	$350,000,000	General Catalyst, …	…	3/14/23	14	3
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
Harbinger Health	$140,000,000	n/a	…	9/25/23	25	9
EquipmentShare	$150,000,000	BDT & MSD Partners	…	9/13/23	13	9
PayJoy	$150,000,000	Warburg Pincus	…	9/5/23	5	9
Alto Pharmacy	$120,000,000	n/a	…	9/25/23	25	9
D-Matrix	$110,000,000	Temasek	…	9/6/23	6	9
Inceptive	$100,000,000	NVentures, Andrees…	…	9/7/23	7	9
Vesper Energy	$100,000,000	GCM Grosvenor	…	9/13/23	13	9
Writer	$100,000,000	Iconiq Growth	…	9/18/23	18	9
Pryon	$100,000,000	US Innovative Tech…	…	9/19/23	19	9
Openly	$100,000,000	Eden Global Partners	…	9/21/23	21	9

342 rows (20 shown) 9 columns

Write DataFrame to Lakehouse

```
from deltalake.writer import write_deltalake
import duckdb
import pyarrow
import polars as pl

# Write DataFrame to Lakehouse
write_deltalake(
    table_or_uri=table_path
    ,storage_options=file_system_client
    # ,data=df.to_arrow() # Polars DF
    ,data=df.arrow() # DuckDB (arrow DF)
    ,mode="append" # Supported options: 'append', 'overwrite'
    ,engine="rust"
)
```

DESCRIBE HISTORY

```
from deltalake import DeltaTable
import pandas as pd

# Initialize the DeltaTable
dt = DeltaTable(table_path)

# Retrieve the full history of the DeltaTable
history = dt.history()

# Convert the history list to a pandas DataFrame
history_df = pd.DataFrame(history)

# Parse the timestamp column
history_df['timestamp'] = pd.to_datetime(history_df['timestamp'], unit='ms')

# Display the DataFrame, sorted by version in descending order
display(history_df.sort_values(by='version', ascending=False).head(5))
```

	timestamp	operation	operationParameters	clientVe
0	2024-07-02 19:57:56.736	WRITE	{'mode': 'Overwrite', 'partitionBy': '[”Month”]'}	delta-rs.
1	2024-07-02 19:55:33.151	WRITE	{'mode': 'Append', 'partitionBy': '[”Month”]'}	delta-rs.
2	2024-07-02 19:53:09.165	VACUUM END	{'status': 'COMPLETED'}	NaN
3	2024-07-02 19:53:07.140	VACUUM START	{'retentionCheckEnabled': True, 'defaultRetent...	NaN
4	2024-07-02 19:51:21.406	VACUUM END	{'status': 'COMPLETED'}	NaN

Download Items from Lakehouse (Files/Tables)

```
# Tables
OneLakeUtils.download_from_lakehouse(
    file_system_client=file_system_client
    ,lakehouse_id=lakehouse_id
    # ,target_file_path="Tables/venture_funding_deals" # Single Table
    ,target_file_path="Tables/" # All Tables
)

# Files
```

```
OneLakeUtils.download_from_lakehouse(
    file_system_client=file_system_client
    ,lakehouse_id=lakehouse_id
    # ,target_file_path="Files/Contoso/contoso_sales.csv" # Single File
    # ,target_file_path="Files/Contoso/" # Subfolder
    ,target_file_path="Files/" # All Subfolders & Files
)
```

Delete Items from Lakehouse (Files/Tables)

```
# Tables
OneLakeUtils.delete_file(
    file_system_client=file_system_client
    ,lakehouse_id=lakehouse_id
    # ,lakehouse_dir_path="Tables/venture_funding_deals_delta" # Single Table
    ,lakehouse_dir_path="Tables/" # All Tables
)

# Files
OneLakeUtils.delete_file(
    file_system_client=file_system_client
    ,lakehouse_id=lakehouse_id
    # ,lakehouse_dir_path="Files/Contoso/contoso_sales.csv" # Single File
    # ,lakehouse_dir_path="Files/Contoso" # Subfolder
    ,lakehouse_dir_path="Files/" # All Subfolders & Files
)
```