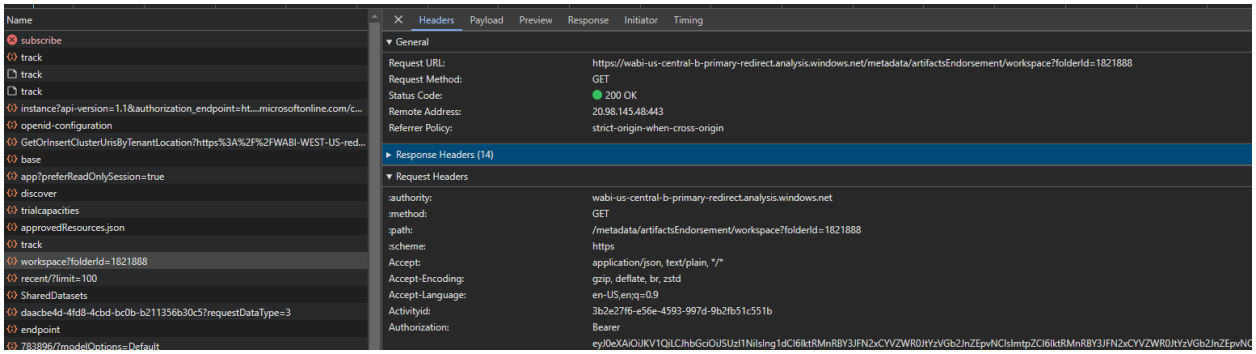
Generic Mirror Mount Creation Steps to follow

Pre-requisites:

- Dev tools to inspect network calls.
- Postman to make backend API calls.
- Microsoft azure storage explorer to upload the source files to mount.

Steps:

1. **Open Fabric App and Monitor Network Calls:**
   Go to the fabric app URL and keep the inspect tool open to trace any network calls.
   https://app.fabric.microsoft.com/
2. **Navigate to the workspace where the mirror artifact is going to be created:**
   Copy the following:
   a) Bearer Token from any call in the network trace (refresh token if it expires)



   b) Workspace id from the URL
   https://app.fabric.microsoft.com/groups/**daacbe4d-4fd8-4cbd-bc0b-b211356b30c5**/list?experience=power-bi

3. **Get the host URL:**

   URL: https://api.powerbi.com/metadata/cluster

   AuthType: *Bearer Token*

   Token: *AAD Token from network trace*

   Response has the ***backendUrl*** which can be used as the host url in next steps.

4. **Create an Artifact :** Create an artifact from using below API call,

URL: https://__HostURL__/metadata/workspaces/__WORKSPACEID__/artifacts

Example host URL: wabi-us-central-b-primary-redirect.analysis.windows.net

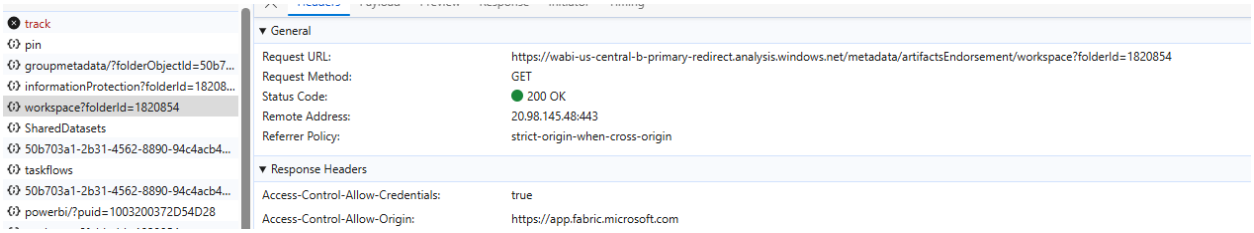
HostURL is the same as backendUrl got from step 3

AuthType: *Bearer Token*

Token: *AAD Token from network trace*

Body:

```
{
    "ArtifactType": "MountedRelationalDatabase",
    "DisplayName": "TestGenericMirrorName"
}
```



From the response save the “objectId”. This is the MountedDatabaseId to be used in step 7.

5. **Verify Artifact Creation:** You can see the created artifact in the UX.

6. **Generate Special User Token:** Make an API call to generate the special user token.

   API - https://__HostURL__/metadata/v201606/generatemwctokenv2

   HostURL is the same as backendUrl from step 3

   AuthType: *Bearer Token*

   Token: *AAD Token from network trace*

   Payload:

```
{
    "workspaceObjectId": "__workspaceID__",
     "workloadType": "DMS"
```

```
}
```

From the response keep a note of the following to be used in step 7

a. Token – This is the special user token to be used
b. TargetUriHost – This is the host url to be used
c. Capacity Object Id – This is the capacity id to be used

7. **Upsert Mounting Config:**

API - https://__HOSTURL__/webapi/capacities/__CAPACITYID__/workloads/DMS/DmsService/automatic/datamarts/__MOUNTEDDATABASEID__/upsertmountingconfig

HostURL is the TargetUriHost from step 6
MOUNTEDDATABASEID is the ObjectId from step 4

Payload:

```
{

    "extendedProperties": {
      "sourceType": " GenericMirror",
       "targetStatus": "Running",
      },
"replicatorPayload": "{\"properties\":{\"source\":
{\"type\":\"GenericMirror\"},\"target\":{\"type\":\"MountedRelationalDatabase\"
,\"typeProperties\":{\"format\":\"Delta\"}}}}"

}
```

Header:
x-ms-workload-resource-moniker: __WORKSPACEID__

Authorization:

AuthType: *API Key*

Token: *MwcToken {{token from step 6}}*

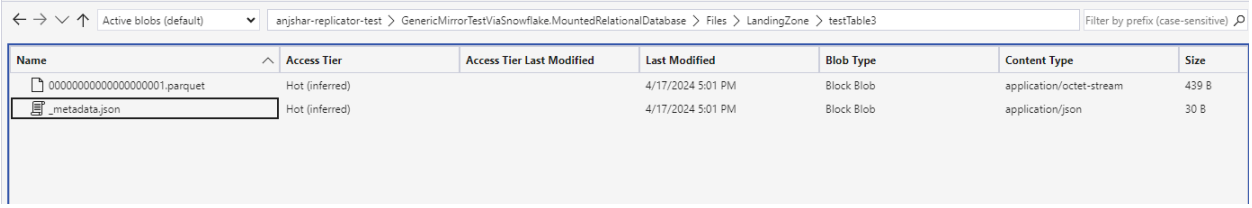
8. **Setup Connection in Azure Storage Explorer :** Next go to azure storage explorer and set up connection to your mounted relational database.

adlsgen2 blob URL: https://onelake.dfs.fabric.microsoft.com/__workspacename__

9. **Create Landing Zone Folder :** Go to the mount you created and create a LandingZone folder under Files folder,

   `TestGenericMirrorName.MountedRelationalDatabase/Files`

10. **Upload Test Data :** Let's upload the test table folder with initial snapshot data and _metadata.json files such as below,



11. **Verify Mirrored Table :** Verify the mirrored table in the UX.