REST: Refreshing vended credentials

Author: Eduard Tudenhöfner <etudenhoefner@apache.org>
Pull requests: REST Spec changes / Impl to refresh vended GCS credentials

Motivation

When a table is loaded, the OpenAPI REST spec supports passing vended credentials to the client, which are used to configure the respective FileIO implementation in order to be able to access a table's data.

However, there is currently no mechanism available that would allow refreshing these vended credentials automatically, as the only available option right now would be to reload the entire table. Reloading the entire table can be expensive if that table has lots of entries in **metadata-log / snapshots / snapshot_log / schemas**.

This proposal aims at providing a mechanism to automatically refresh vended credentials in a way that doesn't require reloading the entire table.

Table credentials endpoint

A new endpoint is introduced to fetch vended credentials for a given table. The endpoint's path is defined as following:
/v1/{prefix}/namespaces/{namespace}/tables/{table}/credentials

This endpoint allows fetching valid vended credentials for the given table.
A server can communicate this endpoint with all placeholders replaced to a client via the **config** of **LoadTableResult**.
A client needs to configure the respective FileIO instance if it receives a credentials endpoint in the **config** of **LoadTableResult** in a way that enables the vended credentials to be automatically refreshed before they expire.

Refreshing vended credentials

Each storage provider has their own mechanism for how to plug in a handler that allows refreshing vended credentials.

For example, GCS supports configuring credentials with a refresh handler as shown below:

```
StorageOptions.Builder builder = StorageOptions.newBuilder();
// other settings ...

// set when gcs.oauth2.refresh-credentials-endpoint property is defined
OAuth2CredentialsWithRefresh credentials =
    OAuth2CredentialsWithRefresh.newBuilder()
                            .setAccessToken(accessToken)
                            .setRefreshHandler(<custom refresh handler>)
                            .build();

builder.setCredentials(credentials);
```

The custom refresh handler will be set in **GCSFileIO** if the server sends the credentials endpoint via the **gcs.oauth2.refresh-credentials-endpoint** configuration property.

The custom refresh handler implements the interface **OAuth2CredentialsWithRefresh.OAuth2RefreshHandler**, which requires implementing **public abstract AccessToken refreshAccessToken()**.
The custom refresh handler's **refreshAccessToken()** calls the respective credentials endpoint and fetches valid vended credentials with a configured expiration time.
The Google library inspects the configured token expiration and calls **refreshAccessToken()** before the token is about to expire, thus enabling automatic refreshes of vended credentials.

JSON representation

A new **LoadCredentialsResponse** with a required **storage-credentials** field is added. Below is an example that shows the **storage-credentials** field with S3 and GCS credentials inside **LoadCredentialsResponse.**

```
// LoadCredentialsResponse
{
  "storage-credentials": [
    {
      "prefix": "s3://custom-uri",
      "config": {
        "s3.access-key-id": "keyId",
        "s3.secret-access-key": "accessKey",
        "s3.session-token": "sessionToken"
      }
    },
    {
      "prefix": "gs://custom-uri",
      "config": {
        "gcs.oauth2.token": "gcsToken1",
        "gcs.oauth2.token-expires-at": "1000"
      }
    },
    {
      "prefix": "gs",
      "config": {
        "gcs.oauth2.token": "gcsToken2",
        "gcs.oauth2.token-expires-at": "2000"
      }
    }
  ]
}
```

Alternative approaches

An alternative to introducing a new credentials endpoint would be to modify the existing endpoint for loading a table (**/v1/{prefix}/namespaces/{namespace}/tables/{table}**) in a way that would allow this endpoint to only return new credentials without the actual table metadata.
However, this would be a breaking change, since **metadata** in **LoadTableResult** is a required field.