**Google Drive File Upload & Download**

The main moto of this two step is uploading & downloading any types of files ( binary files ) from and to, To google drive by using google Oauth 2.0 for accessing bearer access token for sending along with request. While downloading it will convert to bytes and to base64 encoder and while upload decoded that encoded data and send. The feature of this step is upload and download the files from S3 and One Drive and Box api. This will need only binary data for exchanging the files.

And i created total 2 steps they are………..

**For Google Drive File Upload**

1. GoogleDriveFileUploadStep
2. GoogleDriveFileUploadStepExecutor

**1. GoogleDriveFileUploadStep POJO Class: -**

**I** have created pojo with some properties the properties are follows……………..

|  |  |  |
| --- | --- | --- |
| **Sl No.** | **Property** | **Explanation** |
| 1. | **file\_name** | This property is used to set the name of file |
| 2. | **file\_data** | This property accept binary data to upload |
| 3. | **folder\_id** | This property is used to upload the file to specific folder id or it’s optional |
| 4. | **description** | This property is about setting the description of file |
| 5. | **file\_access** | This property is used to enable or disable the permissions to uploaded file like download and view for shared file i.e. true or false |
| 6. | **access\_token** | This property is used to get the access token from google Oauth generated and we are sending that token along with our url as Authorization header.  Ex. Beaer “access\_token” |
| 7. | **output\_variable** | This property is used to hold the generated response for accessing or displaying the result to next step |
| 8. | **value\_type** | This property type of Handlebars or FTL |
|  |  |  |

2. And created a property in StepType.class as a *GOOGLE\_DRIVE\_FILE\_UPLOAD*

3. And added executor class to Step class

2. **GoogleDriveFileUploadStepExecutor : -**

**Step by step process……………………..follows**

* Created a url for hitting.
* Created RequestBody variable for metadata
* And created if condition for folder id is empty then pass the metadata without folder id else passing metadata with folder id
* Decoded the received file data byte[] fileBytes = Base64.*decodeBase64*(step.getFile\_data());
* And created FormDataPart Builder for adding metadata along with file content
* And made a post call with bearer token
* And executed that request using OkHttpClient Ex. Response response = okHttpClient.newCall(request).execute();

**For Google Drive File Download**

1. GoogleDriveFileDownloadStep
2. GoogleDriveFileDownloadStepExecutor

**1. GoogleDriveFileDownloadStep POJO Class: -**

|  |  |  |
| --- | --- | --- |
| **Sl No.** | **Property** | **Explanation** |
| 1 | **file\_id** | This property is used to get the file from the google drive by using file id |
| 2 | **access\_token** | This property is used to get the access token and attach along with  Request header as a bearer “Authorization” token |
| 3 | **output\_variable** | This property is used to hold the generated response for accessing or displaying the result to next step |
| 4 | **value\_type** | This property type of Handlebars or FTL |

**2. GoogleDriveFileDownloadStepExecutor : -**

* **Url in String.format()**
* **ByteArrayOutputStream for writing bytes to file**
* **InputStream is for reading the url by using openStram()**
* **Creating request object and attach get() method and authorization**
* **Reads all the bytes and storing in ByteArrayOutputStream**
* **Encoded the bytes to byteArray**
* **And adding that string encoded file to output variable with UTF Charset**
* **And the byte array is as pass to next step.**

**Providers or Features**

Drive storage is a service where data is remotely maintained, managed, and backed up. The service is available to users over a network, which is usually the internet. It allows the user to store files online so that the user can access them from any location via the internet. The provider company makes them available to the user online by keeping the uploaded files on an external server. This gives companies using cloud storage services ease and convenience, but can potentially be costly. Users should also be aware that backing up their data is still required when using cloud storage services, because recovering data from cloud storage is much slower than local backup.