

My Drive Demo

Database: Google BigQuery (Location: Asia (Delhi))

Tables:

- User_table
- files_table
- user_files_table

File are uploaded on Google cloud Storage(GCS buckets).

Bucket Name: mydrive_99100/DriveUser

API's:

/insertUser

POST http://localhost:8080/insertUser

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "name": "Ramesh",
3   "email": "Ramesh@gmail.com",
4   "password": "rk123"
5 }
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize Text

```
1 Ramesh@gmail.com is added as a user, Now he can Upload flies to his drive
```

/getUsers

GET http://localhost:8080/getUsers

Params Authorization Headers (6) **Body** Pre-request Script

Query Params

KEY
Key

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "user_id": 14,
4     "name": "Ramesh",
5     "email": "Ramesh@gmail.com",
6     "password": "rk123"
7   },
8   {
9     "user_id": 11,
10    "name": "shubham",
11    "email": "shubham@gmail.com",
12    "password": "shubham123"
13  }
14 ]
```

/uploadFile (response when user is authenticated and file is uploaded)

The screenshot shows a REST client interface with a POST request to `http://localhost:8080/uploadFile`. The request body is a JSON object with the following fields: `email` (Ramesh@gmail.com), `password` (rk123), `fileLocation` (D:/source/photo.jpg), and `fileName` (ramesh2.jpg). The response body is also a JSON object with the following fields: `fileLocation` (File at D:/source/photo.jpg is uploaded to bucket mydrive_99100/DriveUser/ramesh2.jpg), `file_upload_status` (Rows successfully inserted in user_files_table), and `authorisation_status` (user is authorised).

```
POST http://localhost:8080/uploadFile
```

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▾

```
1 {
2   "email": "Ramesh@gmail.com",
3   "password": "rk123",
4   "fileLocation": "D:/source/photo.jpg",
5   "fileName": "ramesh2.jpg"
6 }
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ▾

```
1 {
2   "fileLocation": "File at D:/source/photo.jpg is uploaded to bucket mydrive_99100/DriveUser/ramesh2.jpg",
3   "file_upload_status": "Rows successfully inserted in user_files_table",
4   "authorisation_status": "user is authorised"
5 }
```

/uploadFile (response when user is authenticated but has already uploaded the same file in past)

The screenshot shows a REST client interface with a POST request to `http://localhost:8080/uploadFile`. The request body is a JSON object with the following fields: `email` (Ramesh@gmail.com), `password` (k123), `fileLocation` (D:/source/photo.jpg), and `fileName` (ramesh1.jpg). The response body is a JSON object with the following fields: `fileLocation` (user has already uploaded this file, try uploading a different file), `file_upload_status` (invalid), and `authorisation_status` (user is authorised).

```
POST http://localhost:8080/uploadFile
```

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▾

```
1 {
2   "email": "Ramesh@gmail.com",
3   "password": "k123",
4   "fileLocation": "D:/source/photo.jpg",
5   "fileName": "ramesh1.jpg"
6 }
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ▾

```
1 {
2   "fileLocation": "user has already uploaded this file, try uploading a different file",
3   "file_upload_status": "invalid",
4   "authorisation_status": "user is authorised"
5 }
```

/uploadFile (response when user is not authenticated)

POST

http://localhost:8080/uploadFile

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   ... "email": "Ramesh@gmail.com",
3   ... "password": "k123",
4   ... "fileLocation": "D:/source/photo.jpg",
5   ... "fileName": "ramesh2.jpg"
6 }
```

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "fileLocation": "",
3   "file_upload_status": "invalid",
4   "authorisation_status": "user not Authorised, check emailid or password"
5 }
```

Screenshots DB and Storage Bucket:

User_table

QUERY

SCHEMA

DETAILS

PREVIEW

Filter

Enter property name or value

Field name	Type	Mode
user_id	INTEGER	REQUIRED
user_name	STRING	NULLABLE
email	STRING	REQUIRED
password	STRING	NULLABLE

 files_table  QUERY 



SCHEMA

DETAILS

PREVIEW

 Filter Enter property name or value

Field name	Type	Mode
file_ID	INTEGER	NULLABLE
file_name	STRING	NULLABLE
location	STRING	NULLABLE

 user_files_table  QUERY

SCHEMA

DETAILS

PREVIEW

 Filter Enter property name or value

Field name	Type	Mode
user_id	INTEGER	REQUIRED
file_id	INTEGER	NULLABLE

GCS Bucket:


OBJECTS	CONFIGURATION	PERMISSION	PROTECTION	LIFECYCLE
---------	---------------	------------	------------	-----------

Buckets > mydrive_99100 > DriveUser 


[UPLOAD FILES](#) [UPLOAD FOLDER](#) [CREATE FOLDER](#) [MANAGE HOLDS](#) [DOWNLOAD](#)

















[DELETE](#)

Filter by name prefix only ▼

 Filter Filter objects and folders

☐ Show deleted data



<input type="checkbox"/>	Name	Size	Type	Created ?	
<input type="checkbox"/>	 SS.jpg	55.7 KB	application/octet-stream	9 May 20...	 
<input type="checkbox"/>	 archit1.jpg	55.7 KB	application/octet-stream	9 May 20...	 
<input type="checkbox"/>	 courses.PNG	61.3 KB	image/png	9 May 20...	 
<input type="checkbox"/>	 k.jpg	55.7 KB	application/octet-stream	8 May 20...	 
<input type="checkbox"/>	 naman.jpg	55.7 KB	application/octet-stream	9 May 20...	 
<input type="checkbox"/>	 naman1.jpg	55.7 KB	application/octet-stream	9 May 20...	