

Activity 5 lab

| | |
|--|----------------------------|
| <input checked="" type="checkbox"/> Is study | <input type="checkbox"/> |
| ➔ Project | Cloud comp |

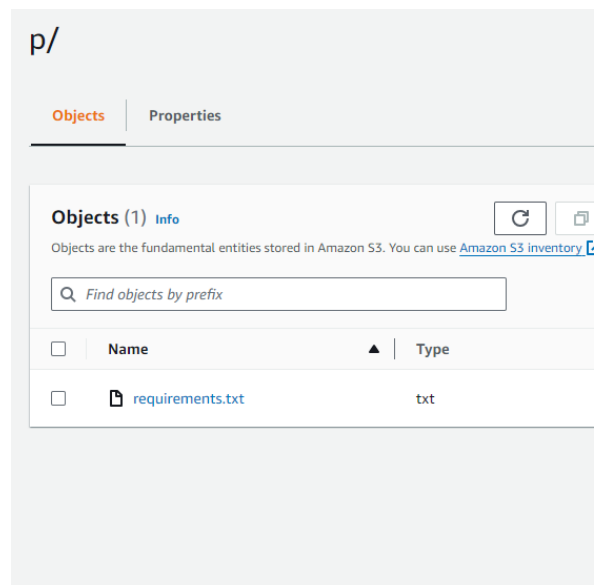
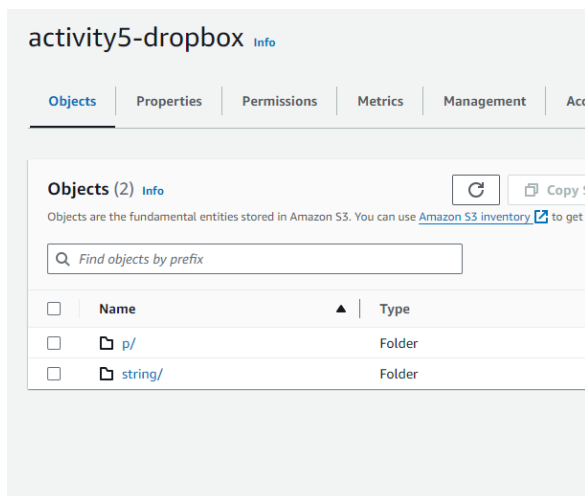
GitHub - chanakorn-aramsak/myDropbox: myDropbox is a Python-based command-line application that allows you to securely manage your files through an API. It provides functionalities for: - Creating folders and files in a secure manner

<https://github.com/chanakorn-aramsak/myDropbox/tree/main>

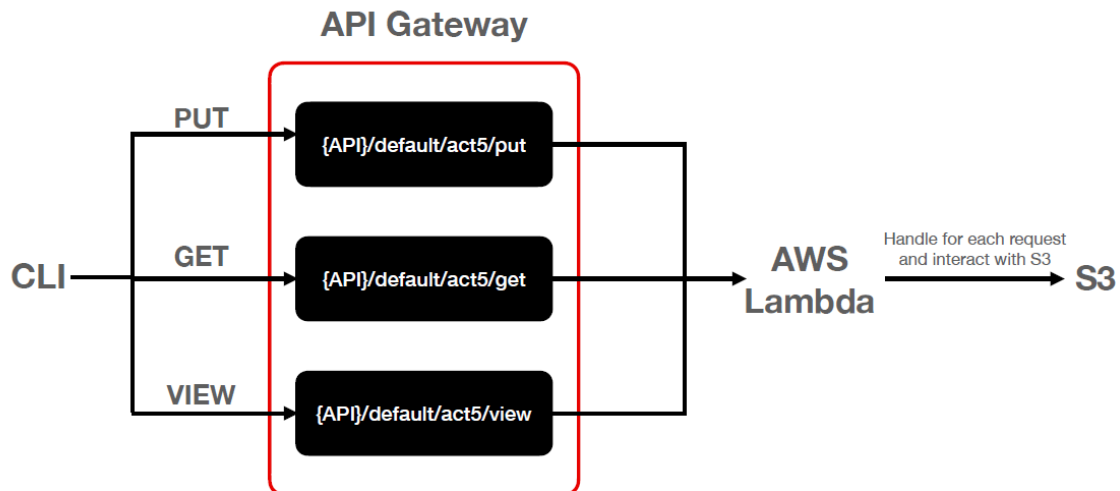
S3 design

ใช้วิธีการจัดการเมื่อมี User หลายคนโดยการสร้าง Folder แยกสำหรับแต่ละ user

- ซึ่ง user ถูกแบ่งให้ distinct ด้วย email ในอนาคตก็จะสร้างชื่อ folder ตาม user email



Overview system



Readme for myDropboxClient:

Description:

myDropbox is a Python-based command-line application that allows you to securely manage your files through an API. It provides functionalities for:

- Creating new user accounts
- Logging in to existing accounts
- Uploading and downloading files
- Viewing a list of your files

Installation:

1. Install required dependencies:

```
pip install -r requirements.txt
```

2. Create a `.env` file in the root directory of the project and add the following line:

```
API_GATEWAY = "https://2rgp0s5019.execute-api.us-west-2.amazonaws.com/default"
```

Usage:

1. Start the application:

```
python myDropboxClient_6330078421.py
```

2. Follow the on-screen instructions to create a new user account, log in, or use available commands:

- `newuser username password password`: Create a new user
- `login username password`: Log in to existing account
- `put filename`: Upload a file
- `get filename username`: Download a file
- `view`: List your files
- `quit`: Exit the application



This is my sample data in S3

[Amazon S3](#) > [Buckets](#) > [activity5-dropbox](#) > [p/](#)

p/

Objects | Properties

Objects (2) [Info](#) ↻ 📄 Copy S3 URI 📄 Copy URL

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket.

| <input type="checkbox"/> | Name | Type | Last modified |
|--------------------------|----------------------------------|------|-------------------------------|
| <input type="checkbox"/> | readme.md | md | February 13, 2021 (UTC+07:00) |
| <input type="checkbox"/> | requirements.txt | txt | February 13, 2021 (UTC+07:00) |

- you can try command to add data into this bucket username = `p`
 - `view` (still can view only username = `p`)
 - `get` `readme.md` `p`
 - `put <filename.txt>` to push data into username = `p`

API documentation for lambda function

API 1: GET /get

Purpose

Downloads a specific file owned by the specified user.

Request Parameters

| Parameter | Required | Description |
|-----------|----------|-----------------------------------|
| owner | Yes | The owner of the file. |
| file_name | Yes | The name of the file to download. |

Success Response

The content of the requested file, with the Content-Type header set to the appropriate file type.

Error Response

- Status code 404: File not found.

- Status code 403: Unauthorized access.
- Other status codes as appropriate.

Example Usage

```
curl --request GET \\  
  --url "https://2rgp0s5019.execute-api.us-west-2.amazonaws.com/default/act5/api/v1/get") \\  
  --header 'Content-Type: application/json' \\  
  --data '{  
    "owner": "p",  
    "file_name": "my_image.jpg"  
  }'
```

Expected response

200 OK

1.25 s

100 B

14 Minutes Ago ▼

Preview ▼

Headers

Cookies

Timeline

```
1 {  
2   "file_url": "https://activity5-dropbox.s3.amazonaws.com/string/string.jpg?  
    AWSAccessKeyId=ASTA2UC3FXIXQMA5FEYRN&Signature=9lrvjK68eyJUHIIISUdgPawxXo3gD&x-amz-security-  
token=IQOjb3jPZ2luXV2Ejr%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FwEacXvZLXd1c3QtMiHMEUCI0DuP%2BxmYU7jWsgHjkndz7  
uw5PFPKB8VVKJONECve03CsWiGJxzERSsQd8qg9Dbg3dhdfmbgzhdTds5siQqwffKez%2F14q3AIICxAAGgw3MzAzMzU2HzY5NTYiDDMG  
bJJYrQKQZCWxpCc5Ampd8HQepES73ddVnH3xlXe%2BeY4o0ubPPN0k3np0BJ1H5cgYph9OUT1nTR2hrxbvzT4zX1clrcVmBGCCZFscow%  
2BLJKKAfVIORQooXz28hf32BBHwifGwtUqkgRfBFmowDlpelIfyIAPERVSOWmcrtwyUml%2BDJ9JLYFcAXm%2BndRfwECtctr1b  
1c3NhvzQ5RBqb2dbY5YLEGrksfwl7UX0%2BVph7JNNbIpeUxP67snCFfWAIKVZKPvUlZNfe9chTOEF3Sis407Ry2DUUeYvZJDkfYA5d  
cnuU48PJRWrkThAe9suwnRKdiMoZ5do1pfkbKU99URN3F5KD1zwerCS856hv3qZjhGhk%2FpgmXHusxEHuZkMEtkv1909Iir4VeG0zG%2  
F1RAJ2xV1IOsxOG6tViPDjhftid195ZcYZY1vscvgw206srgY6ngFfdwEq%2B1NxPAJPLZKsyf1K%2F7ZjxdLQg9Oas7LYRSWrz17%  
2BRMJ0%2FnhzTTTTFB0Gb%2BnLSy5xAKUFZoDdzaoqE%2Fql%2BMbwj3Z2FumaFnVBn%2Fvmaju%2FK705OKUubZEm54bmCxjgwsW3%2  
BoD6u%2BNNSHZDe%2B7I2vsRo9dwRv4TstriJT8Br1cteVPS%2BEBrSRK2N1ZmAhd%22BMtEuZ9cECha1an2ekTg%3D%3D&Expires=  
1707820568"
```

API 2: GET /view

Purpose

Lists all files owned by the specified user.

Request Parameters

| Parameter | Required | Description |
|-----------|----------|---------------------------------|
| owner | Yes | The owner of the files to list. |

Success Response

An array of JSON objects, each containing:

- `owner`: The owner of the file.
- `file_name`: The name of the file.
- `file_size`: The size of the file (in bytes).

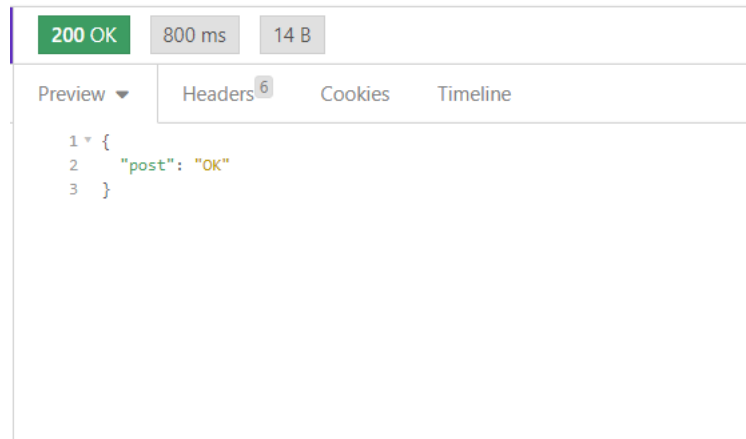
Error Response

- Status code 403: Unauthorized access.
- Other status codes as appropriate.

Example Usage

```
curl --request GET \\  
  --url "https://2rgp0s5019.execute-api.us-west-2.amazonaws.com/default/act5/api/v1/view" \\  
  --header 'Content-Type: application/json' \\  
  --data '{  
    "owner": "p"  
  }'
```

Expected response



API 3: PUT /put

Purpose:

- Uploads a new file owned by the specified user.

Request Parameters:

| Parameter | Required | Description |
|-----------|----------|---|
| owner | Yes | The owner of the file to upload. |
| file_name | Yes | The name of the file to upload, including the file extension. |
| file | Yes | Base64-encoded content of the file to upload. |

Success Response:

- Status code 200: File uploaded successfully.
- Optional response body containing information about the uploaded file (e.g., ID, size).

Error Response:

- Status code 400: Bad request (e.g., invalid parameters, missing data).
- Status code 403: Unauthorized access.
- Status code 413: Payload too large.
- Other status codes as appropriate.

Example Usage:

```
curl --request PUT \\  
  --url "https://2rgp0s5019.execute-api.us-west-2.amazonaws.com/default/act5/api/v1/put" \\  
  --header 'Content-Type: application/json' \\  
  --data '{  
    "owner": "johndoe",
```

```
"file_name": "my_image.jpg",
"file": "iVBORw0KGgoAAAANSUgAAAAEAAAABAQMAAAAL21bKAAAAA1BMVEUAAACnej3aAAAAAXRSTlMAQ
ObYZgAAAApJREFUCNdjYAAAAAIAAeIhvdMAAAASUVORK5CYII="
}'
```

Expected response

200 OK716 ms49 B

Preview

Headers6

Cookies

Timeline

```
1 {
2   "files": [
3     "string",
4     "string.jpg",
5     "string.txt"
6   ]
7 }
```

Prerequisites

- AWS Account
- Amazon S3 bucket
- Amazon Lambda function configured with API Gateway \

