**Effortlessly Fetch and Store the Latest Google Reviews in Azure Blob Storage with Python!**

To implement the process of fetching the top 500 Google reviews of a place and storing them in **Azure Blob Storage** in **JSON format** using **Python**, here's a step-by-step guide with more detail:

**Step 1: Set Up Google API for Google Places Reviews**

Before interacting with the Google Places API to fetch reviews, you need to set up a Google Cloud Project and enable the Places API.

**1.1 Create a Google Cloud Project**

1. Go to the [Google Cloud Console](https://console.cloud.google.com/).
2. Click **Select a Project** at the top.
3. Create a new project by selecting **New Project** and naming it.
4. Note the **Project ID** that will be used for the API.

**1.2 Enable the Google Places API**

1. Navigate to the [Google Places API page](https://console.developers.google.com/apis/library/places-api).
2. Click **Enable** to enable the Places API for your project.
3. Ensure that billing is enabled for the project (Google API requires billing even for free-tier usage).

**1.3 Generate an API Key**

1. Go to the **API & Services** section of your Google Cloud Console.
2. Click on **Credentials**.
3. Click **Create Credentials** and choose **API Key**.
4. Copy the generated API Key (you will use this key to authenticate API requests).

A screenshot of a computer

Description automatically generated

**1.4 Find the Place ID**

The Place ID is required to fetch details, including reviews, for a specific location. You can obtain it in several ways:

1. Use the [Google Place ID Finder](https://developers.google.com/maps/documentation/places/web-service/place-id) to get the Place ID of your location.
2. Alternatively, use a Google search for the place and inspect the URL to get the ID (e.g., placeid=ChIJN1t\_tDeuEmsRUsoyG83frY4).

A screenshot of a map

Description automatically generated

**Step 2: Set Up Azure Blob Storage**

Now, let's set up Azure Blob Storage to store the reviews.

**2.1 Create an Azure Storage Account**

1. Go to the [Azure Portal](https://portal.azure.com/).
2. Create a new **Storage Account**:
   * Click **Create a resource** > **Storage** > **Storage account**.
   * Fill in the required fields (Subscription, Resource Group, and Storage Account Name).
   * Select **StorageV2** for the performance and replication options.

**2.2 Create a Container**

1. After your storage account is created, go to the **Containers** section.
2. Create a new **Container** (e.g., "reviews-container").
3. Set its access level to **Private** or **Blob** depending on whether you want it publicly accessible.

**2.3 Retrieve Connection String**

1. Go to your storage account in the Azure portal.
2. Navigate to **Access keys** under **Security + networking**.
3. Copy **Connection string** (you'll use it to connect your Python script to the storage account).

**Step 3: Install Python Dependencies**

You'll need the following Python libraries:

* **requests** to interact with the Google Places API.
* **azure-storage-blob** to interact with Azure Blob Storage.

Install them via pip:

pip install requests azure-storage-blob

**Step 4: Write Python Code to Fetch Google Reviews and Upload to Azure Blob Storage( example folder structure)**

**google\_reviews\_to\_azure/**

**├── config.py**

**├── fetch\_reviews.py**

**├── upload\_reviews.py**

**├── main.py**

**├── requirements.txt**

**The code is in this location for use.**

**https://github.com/debAzure/goggolereviewtoazureblob**

* 1. **Explanation of Code:**

1. **Fetching Reviews:**  fetch\_google\_reviews() sends a request to the Google Places API using the place ID and API key. It handles pagination and fetches the reviews, limiting it to 500 reviews.
2. **Uploading Reviews to Azure:** upload\_to\_azure\_blob() converts the reviews into JSON format and uploads them to Azure Blob Storage. The reviews are pretty-printed for better readability.
3. **Main Workflow:** The main() function coordinates the process of fetching reviews and uploading them.
4. **config.py**: This file contains configuration settings for the Google API and Azure Storage.
5. **requirements.txt:** This file lists the dependencies for your project.

**4.3 Save and Run the Script:**

1. Save the script as fetch\_google\_reviews\_to\_azure.py.
2. Run the script:
3. python fetch\_google\_reviews\_to\_azure.py

This will:

* Fetch the top 500 reviews from the specified Google Place ID.
* Store those reviews in Azure Blob Storage in the specified container as a google\_reviews.json file.

A screenshot of a computer

Description automatically generated

**Step 5: Verify the Data in Azure Blob Storage**

1. Go to your **Azure Portal** and open your **Storage Account**.
2. Navigate to the **Containers** section and open your **reviews-container**.
3. You should see the google\_reviews.json file.
4. Download or open the file to verify that the reviews were uploaded correctly.

**Step 6: Automate the Script (Optional)**

If you'd like to automate this process to run at regular intervals (e.g., daily), you can:

* Use **Azure Functions** to trigger the script periodically (based on a timer).
* Use **cron jobs** (Linux/Mac) or **Task Scheduler** (Windows) to run the script at set intervals.

**Conclusion**

This detailed guide covers the process from setting up Google API and Azure Blob Storage to writing the Python script for fetching and uploading Google reviews. By following these steps, you should be able to automate the entire process and store reviews in Azure for further analysis or use.