Meowfacts API – Roie Gonen – Assignment Documentation

Description: A simple API that returns random cat facts via GET requests. Meowfacts offers a wide variety of cat-related data in multiple languages.
Objective: To build an integration that updates daily with new facts while ensuring no duplication.

API Capabilities

- **Specific Retrieval:** Fetch a specific fact if the unique ID is known using the *id* parameter.
- **Localization:** Retrieve facts in various languages using the *lang* parameter (refer to the supported language table).
- **Batch Retrieval:** Pull multiple facts in a single request using the *count* parameter; the results returned are randomized.

Key Assumptions

- **Chunk Capability:** It is assumed that the API is capable of pulling the full *fact_count* (the total number of facts available for a specific language) in one single chunk / request.

Implementation Technical Details

1. Dynamic Language Discovery

The script utilizes the */options* endpoint to dynamically discover all available languages supported by the API and their fact count. It iterates through the response to map ISO codes and cumulative *fact_count* values, ensuring that if the API adds new localizations, the script adapts without manual updates.

2. Intelligent Data Collection

- **Cumulative Chunking:** By summing *fact_count* values for duplicate ISO codes (e.g., combining *esp-es* and *esp-mx*), the script optimizes the *count* parameter to request the maximum potential pool of unique facts in a single call.
- **Retry Logic:** The *fetch_facts* method is configured with up to 5 iterations per language to handle potential transient network failures.

3. Data Integrity & Deduplication

To meet the requirement for a clean dataset for analysts:

- **Existing Data Loading:** The script loads the current master file before processing to compare new findings against historical data.
- **Set Comparison:** Python's *set* logic is applied to merge new raw data with existing facts, effectively stripping out duplicates.

4. Error Handling

- **Contextual Logging:** Errors encountered during requests are stored in a list and reported as a summary at the end of the execution.
- **Early Success Break:** If a request is successful, the loop for that language terminates immediately to save resources and avoid redundant calls.

Execution Instructions

1. **Dependencies:** Ensure *requests* library is installed.
2. **Output:** Upon successful completion, a *.json* file named *meowfacts_master_dataset.json* is generated / updated with the latest unique facts categorized by ISO language code.

Script

- File: meowfacts_api.py