Canvas API Academic Data Collector (V1.0)

Project Overview

In this project, we use the Canvas API to collect student academic performance data and integrate it with Notion databases for easy access and analysis.

We focus on collecting and processing:

1. ALL student's currently enrolled courses' scores
2. Processing the data for direct integration with Notion databases
3. Managing the "Is Latest Batch" flag to easily filter the most recent data
4. (Planned) ALL student's currently enrolled courses' assignment status

Dataset: Scores

The dataset includes the following information:

- Student Name: Directly from Canvas API
- Student Chinese Name: Mapping from student credentials
- Student Preferred English Name: Mapping from student credentials
- Course Name: Directly from Canvas API
- Course Name in Chinese: Mapping from Course Name
- Score: Directly from Canvas API
- Grade: Simple mapping: 90-100 A, 80-89 B, 70-79 C, 60-69 D, 0-59 F
- Fetch Time: When the data is fetched (YYYY-MM-DD HH:MM:SS)
- Update Batch: A unique identifier for each data collection batch
- Is Latest Batch: A flag indicating if the record is from the most recent update

We ALWAYS append new data to the existing dataset and manage historical records through the batch flags.

Dataset 2: Assignment Status (Planned for future releases)

The dataset will include the following information:

- Student Name: Directly from Canvas API
- Student Chinese Name: Mapping from Student Name
- Student Preferred English Name: Mapping from Student Name
- Course Name: Directly from Canvas API
- Course Name in Chinese: Mapping from Course Name
- Course Academic Support: Mapping from Academic Support Mapping
- Assignment Name: Directly from Canvas API
- Assignment ID: Directly from Canvas API
- Assignment Due Date: Directly from Canvas API
- Assignment Submission Time: Directly from Canvas API, May be date data or Not submitted
- Assignment Full Mark: Directly from Canvas API
- Assignment Student Score: Directly from Canvas API
- Assignment Status: Directly from Canvas API
- Fetch Time: When the data is fetched. YYYY-MM-DD HH:MM:SS

The planned data update strategy for assignments:

1. If the assignment is new, we will append the data to the existing dataset.
2. If the assignment is not new, we will only update the assignment data as in: fetch time, assignment status, assignment student score.
   This will ensure there is only one record for each assignment and avoids duplicate data.

Installation & Setup

1. Ensure Python 3.8+ is installed on your system
2. Clone this repository
3. Install required dependencies:

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

4. Configure your Canvas API credentials in `credentials.json` (see Security Considerations section)

Project Structure

```
canvas_api/
├── main.py                    # Main entry point
├── config.py                  # Configuration settings and mappings
├── update_student_info.py     # Utility for updating student information in Notion data
├── data_collectors/
│   ├── __init__.py
│   └── grades.py              # Collects course grades
├── notion_processor/
│   ├── __init__.py
│   ├── notion_main.py         # Main entry point for Notion processing
│   ├── notion_api_reference.md # Reference documentation for Notion API
│   ├── data/                  # Notion-specific data files
│   │   └── notion_grades.csv # Notion-friendly formatted grades
│   └── utils/
│       ├── __init__.py
│       ├── notion_formatter.py # Formats data for Notion compatibility
│       ├── batch_manager.py    # Manages update batches
│       └── notion_api/        # Notion API integration
│           ├── __init__.py
│           ├── client.py      # Notion API client
│           └── config.py      # Notion API configuration
├── utils/
│   ├── __init__.py
│   ├── credential_manager.py # Manages student credentials
│   ├── csv_handler.py         # CSV file operations
│   └── error_handler.py       # Error handling utilities
└── data/                      # Directory for storing data files
    ├── grades.csv             # Raw grades data
    └── assignments.csv        # (Planned) Raw assignments data
```

Key Features (V1.0)

1. **Enhanced Student Name Mapping**:
   - Improved mapping between Canvas API names and credential names
   - Support for English names, Chinese names, and pinyin variations
   - Automatic matching of different name formats (e.g., "Jason Jiang" with "Jiang Chenghao")
2. **Direct Notion Database Integration**:
   - Automatically push data to Notion databases via the Notion API

- Format data in a Notion-friendly structure with students as rows and courses as columns
- Update records with proper type formatting for different Notion property types

3. **"Is Latest Batch" Flag Management**:
   - Each new batch of data is marked with an "Is Latest Batch" flag set to "True"
   - Prior to adding new data, all existing "True" flags are set to "False"
   - This allows easy filtering in Notion to display only the most recent data
4. **Improved Course Management**:
   - Automatic detection and mapping of new courses
   - Proper handling of "未知课程" (unknown courses) with accurate Chinese translations
   - Support for a wide range of course formats and naming conventions
5. **Utility Scripts**:
   - `update_student_info.py`: Standalone script to update student information in the Notion data
   - Batch processing and management utilities

Configuration & Mappings

All configurations and mappings are stored in `config.py`:

1. File paths:
   - GRADES_CSV_PATH
   - NOTION_GRADES_CSV_PATH
   - ASSIGNMENTS_CSV_PATH (Planned)
2. Data Mappings:
   - Student Name to Chinese Name
   - Student Name to Preferred English Name
   - Course Name to Chinese Name
   - Course Name to Academic Support (Planned for assignments)
   - Score to Grade mapping

Data Storage

Data is stored in two main formats:

1. `data/grades.csv`: Contains all collected course grades in long format
   - Format: Student Name, Chinese Name, English Name, Course Name, Chinese Course Name, Score, Grade, Fetch Time

2. `notion_processor/data/notion_grades.csv` : Contains grades data formatted for Notion in wide format
    - Format: Student Name, Chinese Name, English Name, Is Latest Batch, [All Course Columns], Updated Time, Update Batch
3. `data/assignments.csv` : (Planned) Will contain assignment data
    - Format: Student Name, Chinese Name, English Name, Course Name, Chinese Course Name, Academic Support, Assignment Name, Assignment ID, Due Date, Submission Time, Full Mark, Student Score, Status, Fetch Time

Notion Integration

The Notion integration now includes:

1. **Direct API Integration**:
    - Full integration with the Notion API to create and update database records
    - Property mapping based on Notion's requirements
    - Support for various property types (title, rich text, number, select, etc.)
2. **"Is Latest Batch" Flag Management**:
    - Automatically resets existing "True" flags to "False" before adding new data
    - Sets the flag to "True" for all newly added records
    - Enables easy filtering in Notion to display only the most recent data
3. **Comprehensive Error Handling**:
    - Detailed error messages for Notion API failures
    - Graceful handling of rate limits and other API issues

Notion API Reference

A comprehensive Notion API reference document is included in `notion_processor/notion_api_reference.md` , covering:

1. Basic setup and authentication
2. Property types and formats
3. Database querying with filters
4. Record creation and updates
5. Best practices and examples

Error Handling

The application implements comprehensive error handling:

1. API Connection Errors: Display connection failure messages with detailed error information
2. Authentication Errors: Provide clear guidance for API token issues
3. Data Processing Errors: Log specific data processing failures
4. File I/O Errors: Handle CSV file read/write exceptions

All errors are raised with descriptive messages to help troubleshoot issues and logged to `canvas_api.log`.

Security Considerations

For security in this project:

1. Student API credentials are stored in a separate `credentials.json` file outside of version control
2. The credentials file is excluded from version control via .gitignore
3. An example `credentials.json.example` file is provided as a template
4. Environment variables can still be used for global configuration
5. Ensure API tokens have minimal required permissions

To set up your credentials:

1. Copy credentials.json.example to credentials.json:
   ```
   cp credentials.json.example credentials.json
   ```
2. Edit credentials.json with each student's actual Canvas API values
3. Never commit the credentials.json file to version control

The credentials.json file uses a format where each student has their own section:

```
{
  "student1": {
    "api_key": "student1_canvas_api_key",
    "domain": "canvas_domain",
    "user_id": 123456,
    "student_name": "Canvas Display Name",
    "student_chinese_name": "中文名",
    "student_english_name": "Preferred English Name"
  },
  "student2": {
    "api_key": "student2_canvas_api_key",
    "domain": "canvas_domain",
    "user_id": 789012,
    "student_name": "Canvas Display Name",
    "student_chinese_name": "中文名",
    "student_english_name": "Preferred English Name"
  }
}
```

Scheduling & Automation

Data collection can be automated using:

1. Manual execution via `main.py`
2. Scheduled execution using system schedulers:
   - cron (Linux/Mac): `0 */6 * * * cd /path/to/project && python main.py`
   - Task Scheduler (Windows)

Dependencies

Required Python packages:

- requests==2.28.1
- python-dateutil==2.8.2
- pandas==1.5.0
- notion-client==2.0.0

Future Enhancements

- Assignment data collection and processing (Dataset 2)
  - Track submission status, grades, and due dates
  - Identify late or missing assignments
  - Generate reports on assignment completion rates
- Email notification system for grade updates
- Interactive dashboard for data visualization
- Support for additional Canvas API endpoints
- Enhanced filtering and reporting capabilities
- Integration with other platforms and services