

Документация к скрипту "my_csv_processor.py"

Описание

Скрипт "my_csv_processor.py" разработан для автоматической загрузки изображений из CSV-файла и их последующего изменения размера с сохранением пропорций сторон. Скрипт также обрабатывает ошибки при скачивании изображений и предоставляет уведомления о статусе выполнения.

Использование

1. Подготовка окружения

- Убедитесь, что у вас установлены необходимые библиотеки: requests, Pillow (PIL). Их можно установить с помощью pip, выполнив следующую команду:

pip install requests pillow

- Создайте папку "avs" в той же директории, где находится ваш скрипт "my_csv_processor.py". Эта папка будет использоваться для сохранения скачанных и измененных изображений.

2. Подготовка CSV-файла

- Убедитесь, что у вас есть CSV-файл с данными для скачивания изображений. Формат файла должен быть следующим:
 - Первый столбец содержит имя файла (без расширения), которое будет присвоено скачанному изображению.
 - Второй столбец содержит URL-адрес изображения для скачивания.
- Поместите CSV-файл в ту же директорию, где находится ваш скрипт "my_csv_processor.py". Вы должны указать имя этого файла в скрипте.

3. Запуск скрипта

- Запустите скрипт "my_csv_processor.py". Это можно сделать из командной строки следующей командой:

python my_csv_processor.py

-
- Скрипт начнет обработку CSV-файла. Он будет поочередно загружать изображения, изменять их размер и сохранять в папку "avs".
- В процессе выполнения вы увидите уведомления о статусе обработки для каждого изображения. Скрипт также предоставит информацию об ошибках, если таковые возникнут.

4. Отслеживание выполнения

- Скрипт будет выполняться до тех пор, пока все строки в CSV-файле не будут обработаны. После завершения выполнения скрипт выведет сообщение "Job is done!".

5. Логирование ошибок

- Скрипт также поддерживает логирование ошибок. Логи ошибок записываются в файл "error_log.txt", который находится в той же директории, где находится скрипт. В логах содержится информация об ошибках при скачивании и обработке изображений.
- Чтобы изменить настройки логирования, вы можете редактировать функцию logging.basicConfig, включая уровень логирования (level) и формат сообщений.

6. Обработка ошибок

- Скрипт обрабатывает ошибки при скачивании и обработке изображений. В случае ошибки, скрипт попытается скачать изображение снова до 20 раз (максимальное количество попыток указано в max_retry_count). Если после этого изображение по-прежнему не может быть загружено или обработано, выполнение скрипта завершается, и в логах будет уведомление об этом.

7. Пользовательские настройки

- В скрипте можно настроить желаемую директорию для сохранения изображений. Измените значение переменной desired_directory, чтобы указать абсолютный путь к директории, в которой вы хотите сохранять изображения.