Fleetio-Smartcar Integration Setup Guide

Fleetio-Smartcar Integration Tutorial

This tutorial will guide you through setting up the Fleetio-Smartcar Integration, which syncs your vehicle data (such as odometer readings) automatically from Smartcar to Fleetio. This integration keeps vehicle data accurate for maintenance tracking.

Requirements

Smartcar: Ensure your Smartcar account is connected to your vehicle(s), such as those in the Toyota app.

Fleetio: API access for syncing data.

Python: Installed on your system (3.6+).

Git: Recommended for version control.

API Credentials: Generate Smartcar and Fleetio API keys.

Step-by-Step Guide

Step 1 - Clone the Project

Download the Project:

Open your command prompt or terminal.

Run the following command to download the project files:

```
git clone https://github.com/deuber/fleetio-smartcar-integration.git
```

Navigate to the Project Folder:

```
cd fleetio-smartcar-integration
```

Step 2 - Set Up Python Environment

Create a Virtual Environment:

In the terminal, create and activate a virtual environment to keep dependencies separate.

On macOS/Linux:

```
python3 -m venv venv
source venv/bin/activate
```

On Windows:

```
python -m venv venv
venv\Scripts\activate
```

Install Required Packages:

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

Step 3 - Obtaining API Keys

Log into Smartcar:

Create or log into your Smartcar account.

Go to the Developer Dashboard and set up an application.

Copy your Client ID and Client Secret.

Log into Fleetio:

Log into your Fleetio account.

Go to API Settings and retrieve your API Token and Account Token.

Step 4 - Configuring .env for Credentials

Create a .env File:

In the project folder, create a file named .env.

Open this file in a text editor and add the following information:

```
SMARTCAR_CLIENT_ID=your_smartcar_client_id
SMARTCAR_CLIENT_SECRET=your_smartcar_client_secret
FLEETIO_API_TOKEN=your_fleetio_api_token
FLEETIO_ACCOUNT_TOKEN=your_fleetio_account_token
REDIRECT_URI=http://localhost:8000/callback
```

Save and Close the File.

Step 5 - Authorize Smartcar Access to Your Vehicle

Command-Line Authorization: The first time you run smart_fetch.py, it will prompt you to authenticate and authorize access to your vehicle data.

Step 6 - Running the Integration

Run the Script:

In the terminal, run the following command:

```
python smart_fetch.py
```

The script will prompt you to log in to Smartcar, retrieve vehicle data, and sync it with Fleetio.

1. Prompts for Smartcar authentication.
2. Retrieves vehicle and odometer data.
3. Syncs with Fleetio, updating or creating entries.

Automating Daily Execution on macOS

Create the schedule_fetch.py Script:

This script will help you automatically update data daily.

Create a file named schedule_fetch.py and paste the following:

```
import subprocess

import datetime


# Log the execution time

current_time = datetime.datetime.now()

print(f"Running smart_fetch.py at {current_time}")


# Run smart_fetch.py

subprocess.run(["python3", "smart_fetch.py"])
```

Save and Make it Executable:

```
chmod +x schedule_fetch.py
```

Setting Up a Cron Job (for macOS users)

Open Cron Table:

In the terminal, type:

```
    crontab -e
```

Add the Daily Job:

Add the following line to run the script at 8:00 AM every day:

```
0 8 * * * /path/to/your/project/schedule_fetch.py >> /path/to/your/project/fetch.log 2>&1
```

Replace /path/to/your/project/ with the actual path.

Additional Information

Contributing:

Fork the repository on GitHub.

Make your changes and submit a pull request for review.

License: