Documentation for the provided Python script

1. Introduction

This Python script performs web scraping on the Khaadi (random website) to extract information about fabric products, including brand, link, and price. The extracted data is then saved to a CSV file and optionally stored in Firebase.

2. Prerequisites

- Python 3: Ensure that Python 3 is installed on your machine.

- Dependencies: Install the required Python libraries using the following:

```BASH
pip install selenium
pip install lxml
pip install pandas
pip install pywhatkit
pip install firebase-admin
```

-WebDriver: Download the chrome WebDriver from below link and make sure it is in your system PATH.

(https://sites.google.com/chromium.org/driver/)

-Firebase Project: Create a Firebase project and download the service account key JSON file (`key.json`) from Firebase Console.

3. Script Components

3.1. Import Statements

Python

```python
import time
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from lxml import html
import pandas as pd
from datetime import datetime
import pywhatkit
import firebase_admin
from firebase_admin import credentials, firestore
```

3.2. Firebase Initialization

Initialize Firebase with your service account key:

Python

```python
cred = credentials.Certificate('key.json')
firebase_admin.initialize_app(cred, {"databaseURL": "https://your-firebase-project-url"})
```

Replace `'key.json'` with the correct path to your Firebase service account key file and provide the appropriate Firebase project URL.

3.3. Web Scraping Functions

- **`scroll_down(driver)`:** Scrolls down to the bottom of the webpage to load all the content.

- **`scrape_khaadi_data(url)`:** Uses Selenium to scrape brand, link, and price information from the Khaadi website.

- **`save_to_csv_pandas(data, filename_prefix='khaadi_data')`:** Saves the scraped data to a CSV file with a timestamped filename.

- **`send_whatsapp_message(data_filename)`:** Sends the extracted data via WhatsApp using `pywhatkit`.

3.4. Data Storage on Firebase

- **`store_data_on_firebase(data)`:** Stores the scraped data in the Firestore database of your Firebase project.

3.5. Main Execution

```python
url = 'https://pk.khaadi.com/new-in/fabrics/'
khaadi_data_df = scrape_khaadi_data(url)
data_filename = save_to_csv_pandas(khaadi_data_df)
send_whatsapp_message(data_filename)
store_data_on_firebase(khaadi_data_df)
```

The script sets the URL, performs web scraping, saves data to CSV, sends a WhatsApp message, and optionally stores data on Firebase.

4. How to Run

1. Save the script with a **`.py`** extension, e.g., `data-scrap.py`.

2. Install Pyinstaller (create exe file)

```bash
pip install pyinstaller
pyinstaller --onefile data-scrap.py
```

3. Run the script using:

```bash
python data-scrap.py
```

Make sure your machine is connected to the internet, and the Chrome WebDriver is correctly configured.

5. Conclusion

The provided Python script is a versatile tool designed to scrape fabric product data from the Khaadi website. It employs a range of libraries, including Selenium for web scraping, Pandas for data manipulation, PyWhatKit for sending WhatsApp messages, and Firebase for optional data storage.

The script begins by initializing Firebase with the provided service account key and project URL. It then defines functions for web scraping, CSV file creation, WhatsApp messaging, and Firebase data storage.

The web scraping functions utilize Selenium to navigate the Khaadi website, extract brand, link, and price information, and store it in a Pandas DataFrame. The data is subsequently saved to a CSV file with a timestamped filename. Optionally, the script sends the extracted data as a WhatsApp message and stores it in a Firebase Firestore collection.

The main execution section orchestrates the entire process, from web scraping to optional data storage on Firebase. To run the script, users need to ensure Python 3 is installed, dependencies are installed via pip, Chrome WebDriver is available, and the Firebase service account key is correctly configured.

This documentation serves as a comprehensive guide, offering insights into the script's purpose, prerequisites, components, main execution flow, and instructions for running it. Users should refer to this documentation to understand and execute the script effectively, making it a valuable tool for extracting and managing data from the Khaadi website.