The backend of this project is responsible for handling various functionalities related to web scraping and sending notifications using Firebase Cloud Messaging (FCM). This documentation provides an overview of the backend code, explaining its components, routes, and functionalities.

**Table of Contents**

1. Introduction
2. Dependencies
3. Routes and Functionality
   * Search by Word
   * Search by Link
   * Infinite Loop and Notification
4. FCM\_handler Module
5. Conclusion

**Introduction**

The backend of this project is built using the Flask framework, which allows for easy creation of web applications. The backend interacts with external websites through web scraping using the **requests** library and parses HTML content using **BeautifulSoup**. Additionally, it utilizes the **flask\_cors** library to handle cross-origin requests and facilitate communication between the frontend and backend.

**Dependencies**

The backend relies on the following Python libraries and modules:

* **Flask**: A micro web framework used for creating web applications.
* **requests**: A library for making HTTP requests.
* **BeautifulSoup**: A library for parsing HTML and XML documents.
* **flask\_cors**: A Flask extension for handling Cross-Origin Resource Sharing (CORS).
* **firebase\_admin**: A Firebase SDK for Python, used for sending notifications using FCM.

**Routes and Functionality**

**Search by Word**

Route: **/search**

This route is used to perform a search based on a provided keyword and retrieve relevant product information from a specified link. It employs web scraping techniques to extract product names, image links, product links, and prices from the HTML content of the target webpage.

**Search by Link**

Route: **/q**

This route allows querying product information directly using a link. It retrieves details such as the product name, price, and image link from the HTML content of the provided link.

Infinite Loop and Notification

**Push Notification**

Route: **/notify**

This route implements an infinite loop that repeatedly checks for a price drop in a product's webpage. When the price drops below a certain expected amount, it sends a notification using the **FCM\_handler** module.

**FCM\_handler Module**

The **FCM\_handler** module is responsible for sending notifications to user devices using Firebase Cloud Messaging (FCM). It initializes Firebase with a provided service account key and defines a **notify** function for sending notifications. This function takes the notification title, message body, registration tokens of target devices, and optional data payload.

**Conclusion**

The backend of this project serves as the engine that powers the web scraping and notification functionalities. Through various routes, it communicates with the frontend and external websites, providing users with real-time product information and alerts when prices drop. The **FCM\_handler** module enhances user engagement by sending notifications via Firebase Cloud Messaging.

For more detailed information about individual code components, please refer to the source code.