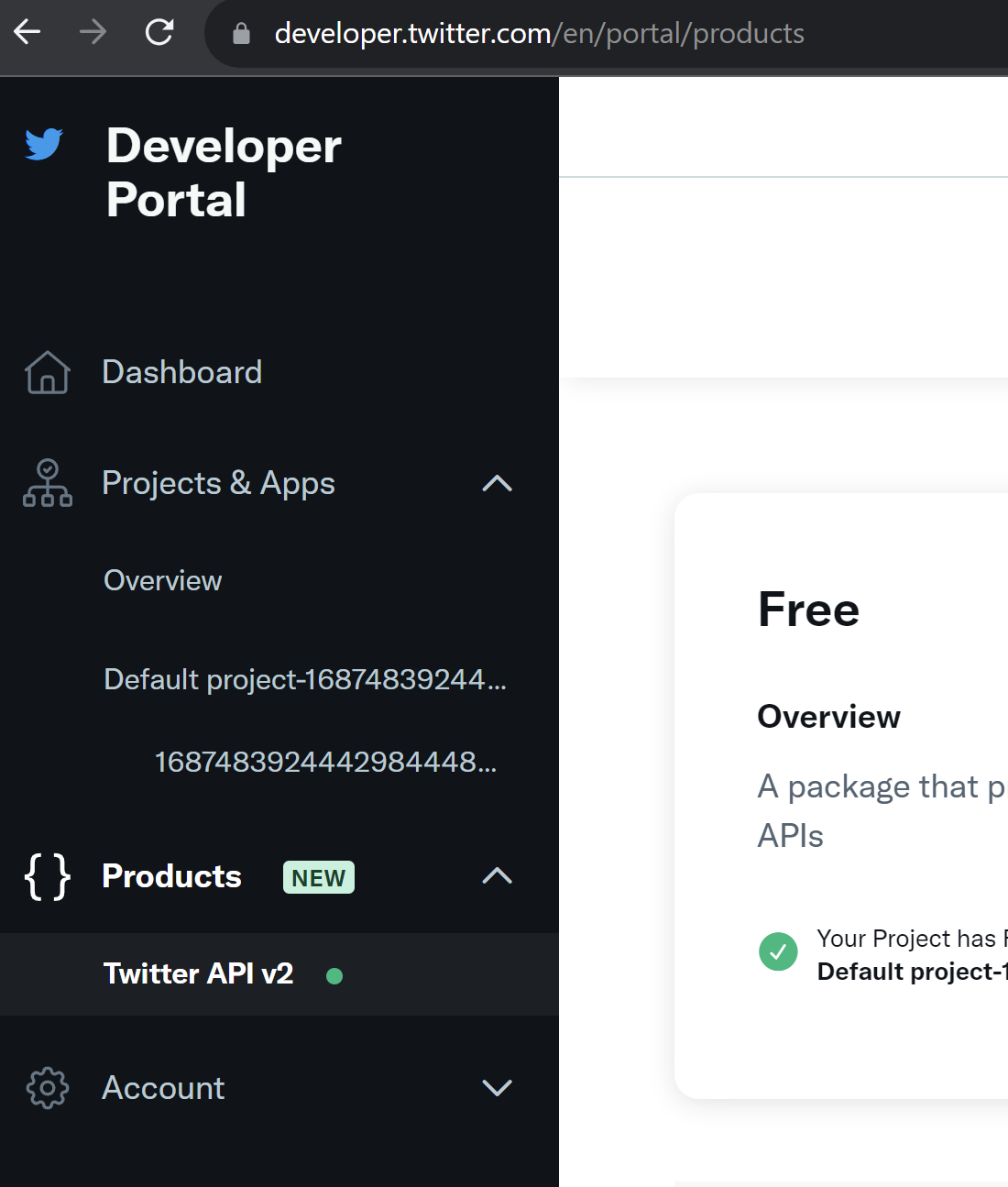
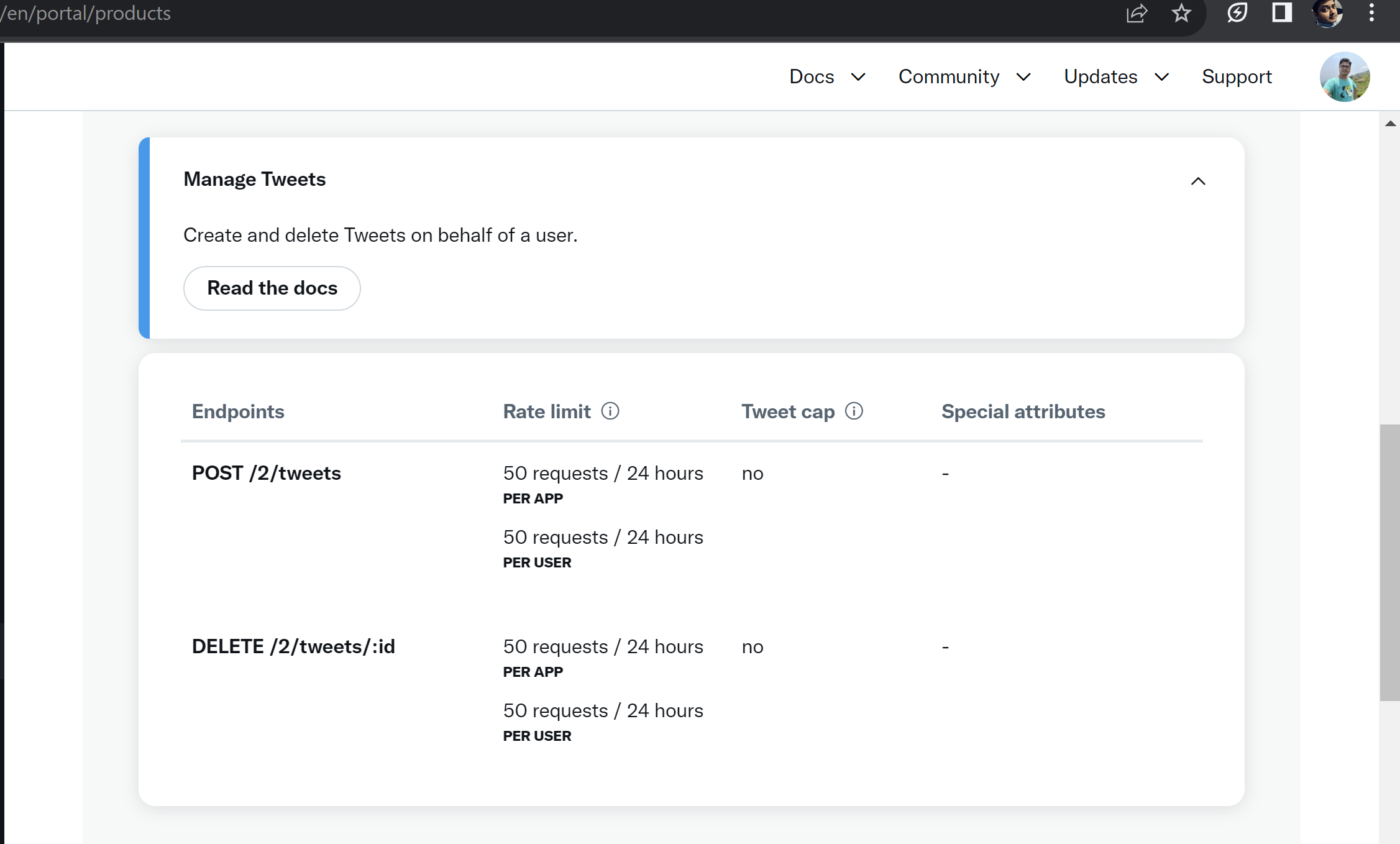
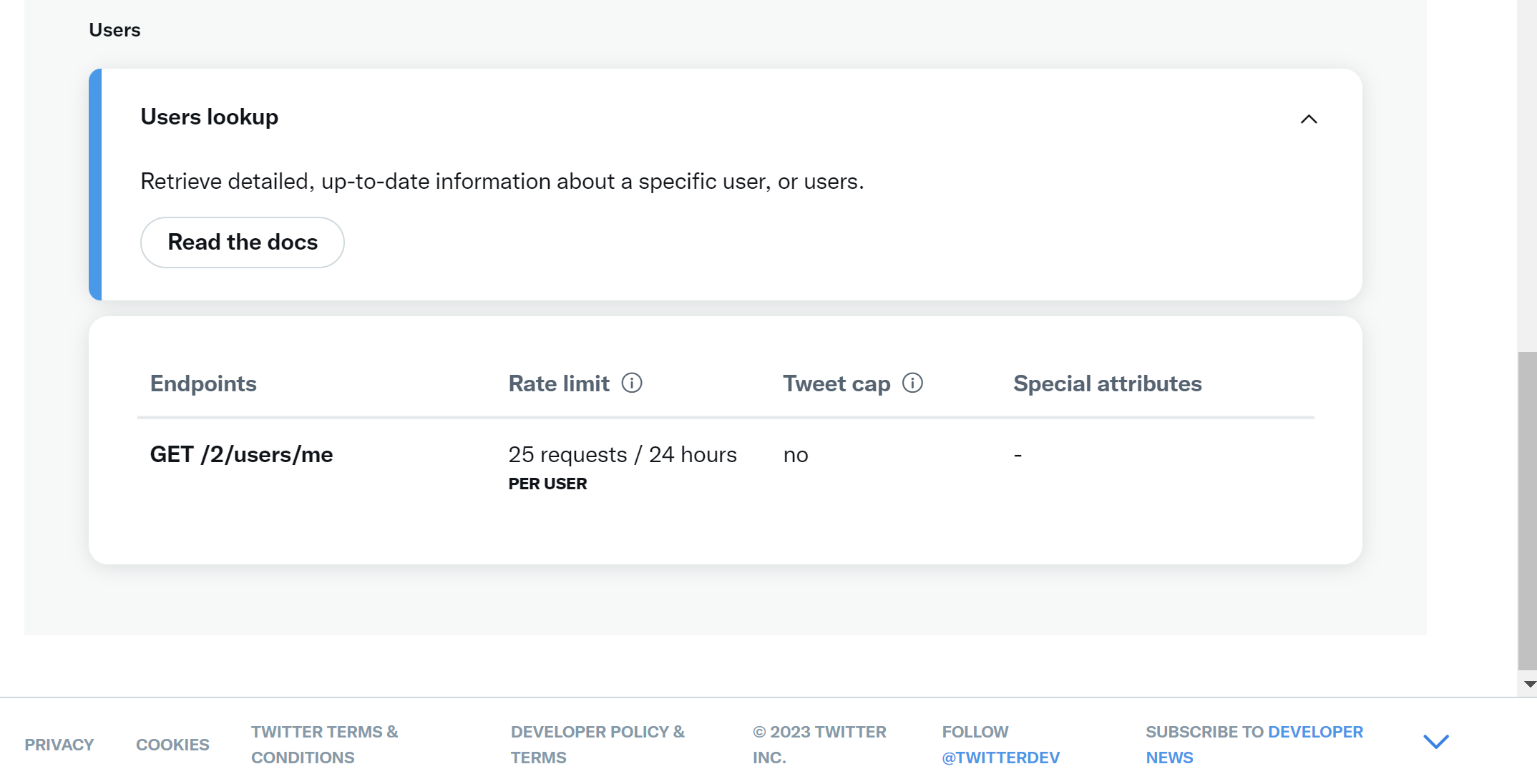
**Case Study Name**: Netflix Titles Streaming and Analytics Platform  
  
**Case Study Description:** Netflix Titles Streaming and Analytics Platform is a configurable solution which utilizes the Kafka components to create a data pipeline which sanitizes the input stream and perform analytics on the Netflix Title’s information along with enrichment of the data.  
  
**Why not Twitter API V1:**  
Twitter API V1 is not working right now, at the time of writing code for this case study. I tried consuming the Twitter API V1 with hosebird client, but it didn’t work out.  
Some more research showed other developers are also facing similar issues with Twitter API V1. This link from twitter community provides information on the same:  
<https://twittercommunity.com/t/v1-1-api-no-longer-working/199890/13>  
  
Twitter API V1 is now no longer showing up in the list of available products in the developer account dashboard:  


**Why not Twitter API V2:**  
Currently the free tier of twitter developer account provides access to these limited API only, which in my opinion is not enough for the asks in the case study:  




That’s where I thought of creating a different API, streaming agents etc all by myself. I used data from Kaggle for Netflix, and stitch up everything myself to satisfy the intent mentioned in this case study.  
  
**Overview of Netflix Titles Streaming and Analytics Platform:**  
  
i) **NetflixTitlesStreamingAgent** in the codebase reads the netflix\_titles.csv file from Kaggle, and produce records to a Kafka topic, **raw-data-topic**, through a custom Kafka producer **NetflixKafkaProducer**. Intentionally, I have added a delay of 1 second, so it makes the topic to receive data from NetflixTitlesStreamingAgent every second continuously.

ii) **NetflixKafkaStreamListener** in the codebase reads from raw-data-topic, applies filter logic, does transformation (stateful as well stateless) on KStream, removes certain keywords and masks them with mask character. And finally, it pushes the sanitized data to **sanitized-data-topic**.

iii) **NetflixKafkaTransformedStreamListener** reads the sanitized data from sanitized-data-topic Kafka topic.

iv) **NetflixTitlesAnalyticsApplication** is the starting point of this application. It stitches all the components together and starts the application. This application keeps running till all the records from the netflix\_titles.csv file has been read by **NetflixTitlesStreamingAgent** and pushed toraw-data-topic.

v) **CustomSerdes** class provide custom serdes used in the application.

vi) **KafkaConfiguration** provides necessary configuration for KafkaStream, KafkaProducer and KafkaConsumer in the application.

vii) Log4j2 has been used in the application.