WikiMedia Stream Monitor Project

Tiffany Cheng

Things done

- Show basic info and some details per entry
- Pagenation
- Filter stream by various aspects
 - Preserve filter control with marks
- Mark seen

Future Improvement

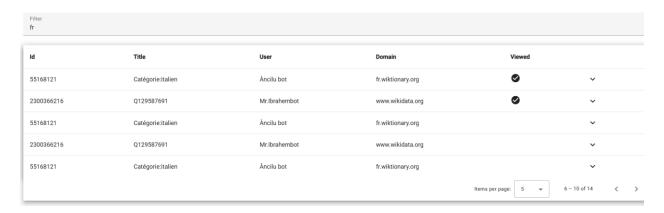
- 1. UI could change to using Real Stream with Observable, Subscribe and Rxjs
 - a. Currently, using python in the backend to fetch 100 records at a time and take down the last id for future requests.
 - b. If change to user the live streaming directly from https://stream.wikimedia.org/v2/stream/recentchange backend might not be required.
- 2. In order to track the changes offline, I will change the backend DB from json file into PostgreSQL DB and use the Python Flask with SQLAlchemy as ODBC. This proper API server could improve the performance and IO with UI for reading and marking entries easily. Also, a separate microservice can keep running in the backend as a listener to the latest changes upstream. In current implementation, the UI does not require many changes, because it's reading and calling the backend API at the moment. A refresh button could be implemented for UI to request a new batch of entries.
- 3. The scaling issue is mainly on fetching of the recent changes from the stream.wikimedia API. Need to be careful about the rate limit and retries.
- 4. For the seen/dealt mark,
 - a. If it's for personal usage, it's better to save it further into cookies.
 - b. If it's sharing between multiple admins/users, we could add as additional attribute in the backend DB to track if someone else has dealt with it.
- 5. Build docker image to avoid local node and Angular dependencies.
- 6. Integrated the data with other APIs liftwing and mediawiki.
- 7. Download the detail of entry as a JSON text file.

Table view of the entry

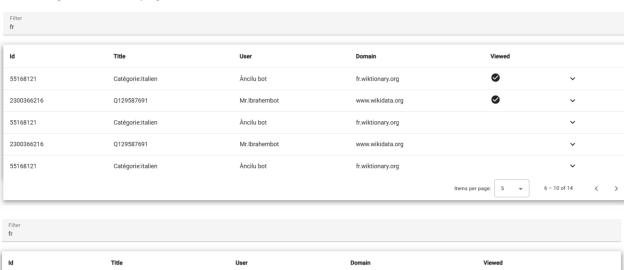
ID, Title, User, Domain, Comment, Timestamp, Go to Page button (Title_url)



Marked Viewed onClick



Preserving the filter on pagination





Change of Tab Icon

