

Cleveland Museum of Art - Google Analytics Test (Ben Luo)

12/12/24

Goal: Get the top artworks based on CLI inputs from Ga4, then get the corresponding metadata for each artwork using the open API from CMA, and display them.

To Run: Run main.py with proper credentials and env set up (requirements.txt)

- `python main.py --start-date 2024-12-01 --end-date 2024-12-12 --top 10`

Process:

1. Go to google cloud console and create a project using the set up (CMA-Analytics-Test)
2. Get credentials from Google Analytics Data API, set up OAuth 2.0 client
 - a. download json credentials
3. Set up a python environment and project, store a client_secrets.json of the above
4. **Blocker 1:** Permission denied when trying to hit Ga4
 - a. Resolved blocker by asking for archived service account key (JSON file) that you can use with the oauth2 library in place of the API key
5. Simple query to see dimension of call (`list_dimensions_and_metrics()`), ctrl+f to find important fields
 - a. **screenPageViews** (want the most viewed) and **pagePath** (includes the accession number)
6. Created new query with a regex to filter on pagePath and use pagination to ensure we get the correct limit of artworks (<https://pythex.org/>)
7. Create CLI tool with table, wanted to get images dict first before displaying the image. Ran out of time, decided to break up code into separate files for easier readability and settle on using a table view

Reflection: If I had more time, I would work on:

- Getting a full display with the thumbnail using a lightweight python gui app. I was able to get the image url, but I just didn't have time to display it.
- Data back to January 1st, 2024 - I wasn't sure what this meant
- I would consider design choices if this app is for internal vs external use, etc.
- Use the API key. I've usually just set permissions for account on AWS console

Result: 2 hours of dev time, 1.5 hour of setup issues with blocker. Apologies for going over, was new to a lot of this tech and got stuck on set up.

- Had fun though, thanks! Really enjoy the open API for CMA art :)

