Collected Knowledge & Wisdom on ### Google OAuth 2.0 — ### Provides: This API provides…

- A secure way for users to provide identifying data without ever disclosing their username and passwords.
- A way to check that users are from an authorized domain.
- A way for users to store documents and other files made on the website into their google drives/emails for easy access.

Pain factor: 3

(0=ezpz…5=nightmare) - You HAVE to do it right because otherwise it's not secure. - Pain factor to do it wrong is very low (1)

Key Provisioning:

- Create an account on Google's Cloud Platform: https://console.cloud.google.com
- Go to APIs and Services and click Credentials. If you haven't yet, you have to fill out the OAuth consent screen.
- Next, click on create credentials and you can make a google oauth client.
- For more specific instructions (there a lot of steps), you can go to this (https://realpython.com/flask-google-login/) tutorial and follow the instructions under the bold header that says Creating a Google Client.

Quotas:

- Default: 100 requests per 100 seconds per user.
- However! There is a maximum of 10 requests to the API per user per second.
- Also, depending on the risk of the application, there is a scope on the app use.

The Good:

- Allows coder to adjust setting on who can log in and from which account.
- Google is secure (much more secure than most high schoolers' self coded login systems would be)
- It is easy to login with google– you don't have to make a separate account for the website that you have to remember.
- No need to implement a "forgot your password" functionality

The Bad:

- If you don't do it correctly or mess up a step, then it could be insecure.
- Lots of steps and handshake orders to understand

The Ugly:

- You can't login to google cloud platform from your stuy.edu account

Location: hhttps://console.cloud.google.com

Accurate as of (last update): 2022-05-27

Contributors:

the two spaces after each name are important!

Eliza Knapp, pd2