Short proposal for Gmail scam detector plug-in

1 Gmail email client is chosen

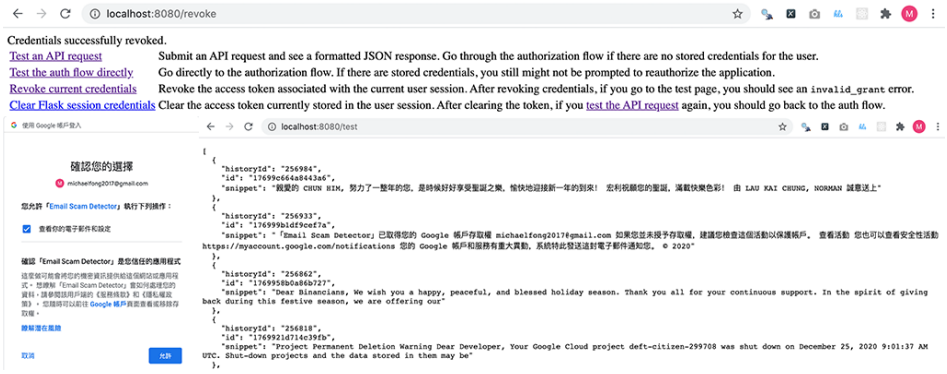
1. **Popularity**: Consider Email Client Market Share in 2020, Gmail email client is 38% and Apple Mail email client is 40%. Besides, Gmail is by far the most popular browser email client.
2. **Public REST API**: Gmail API is public whereas Apple Mail API is private.

2 Other choices

Choice	Reason
Web server application	Zero install (only require a browser); easy development and testing; quick and easy updates
Python as primary language	Great community support for various tasks such as performing Gmail API requests; machine learning libraries are likely used
Spyder as IDE	Split into code cells; debugger; interactive variable explorer

3 Development and deployment

By using Gmail API + Google APIs Client Library for Python + Google's OAuth 2.0 server, I can already run a python development server and read my own emails from a browser. Such web server application will implement the scam detection main logic and can be used standalone.



The next step is to develop a Google Workspace add-on using the necessary Apps Script, giving the UI, behaviours and user interactions of the scam detector within the Gmail email client. This add-on interacts with the web server application via REST API. This add-on is listed, reviewed and published on the Google Workspace Marketplace.

4 Scam detection

Email scams are basically phishing, where fake emails pretend to come from trusted organizations such as banks and online shops. They usually trick victims into going to a website with looks exactly like the real organization's website and then disclosing personal information. Other email scams can direct victim to a fraudulent online shop. **The direct way** of scam detection is that for each incoming email, URLs and, if possible, URLs embedded in images are retrieved and searched in some database using a white list or black list. **An efficient way** is to use machine learning, which integrates URL text features, domain name features and web content features to classify phishing or fraudulent websites. URL text features and domain name features are extracted from words that compose a URL based on query data from search engines like Google. Web content features include the number of repetitive paragraphs for online shops. **A more innovative way** is to use NLP to determine the associated organization (e.g. Bank of East Asia) of the email and search whether the sender of the email is possible (e.g. whether it ends with @hkbea.com).