

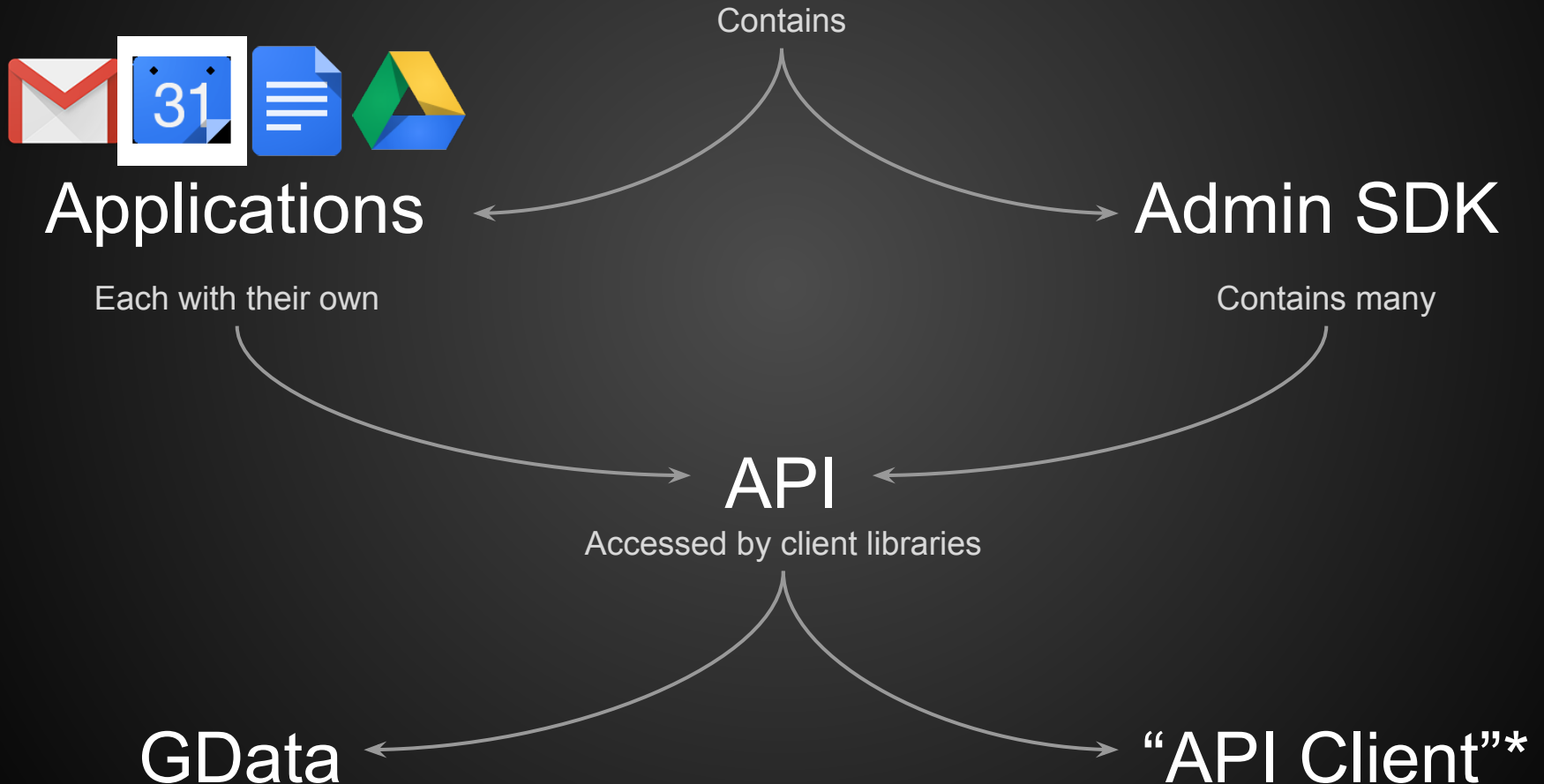


# APIs for Domain Admins

by [warden@geneseo.edu](mailto:warden@geneseo.edu)

# The Lay of the Land

## Google Apps Platform



**\*I do not like the name  
“API Client”**

Imagine if someone tried name a car “The  
Car”...

...Oh, wait.



("Renault5-Le Car" by Jonathan Kellenberg from Seattle, WA - Flickr. Licensed under Creative Commons Attribution 2.0)

# APIs

## Application

Calendar API

Tasks API

Gmail APIs

Contacts API

Apps Activity API

Drive API

Spreadsheets APIs

Sites API

## Admin SDK

Admin Settings API

Calendar Resource API

★ Directory API

Domain Shared Contacts API

Email Audit API

Email Migration API

Email Settings API

Groups Migration API

Groups Settings API

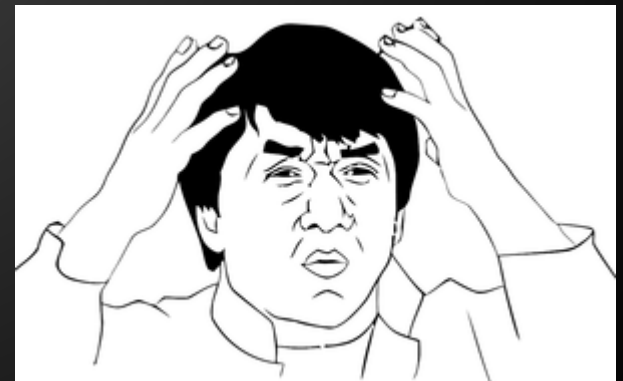
Enterprise License Manager API

★ Reports API

Reseller API

# GData vs “API Client”

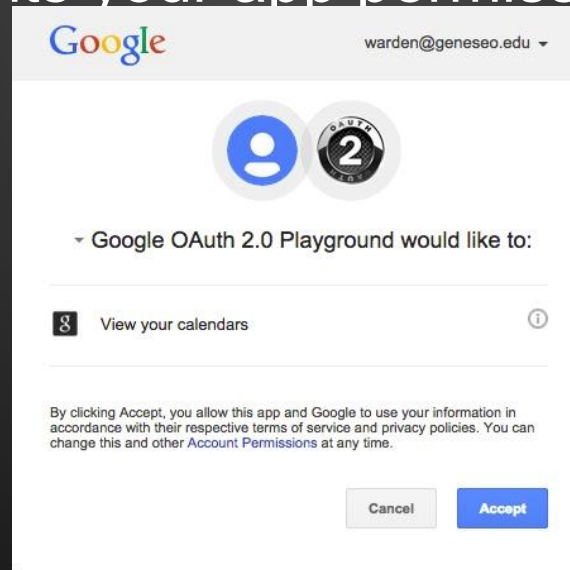
- Generally, “API Client” is newer and preferred
- Some GData APIs have been deprecated and migrated to “API Client”
  - Calendar - Shutdown Nov 17, 2014
  - YouTube Data - Deprecated March 2014
  - Provisioning - Deprecated May 2013
  - Documents List - Deprecated September 2012
- But some things are ONLY available in GData!
  - Contacts
  - Sites
  - Spreadsheets





# Authentication

- OAuth 2.0 is your only option in many APIs
  - 2 Legged OAuth 2.0 (2LO)
    - Administratively access your GApps domain users
    - “domain-wide delegation of authority”
  - 3 Legged OAuth 2.0
    - User grants your app permission at a Google screen

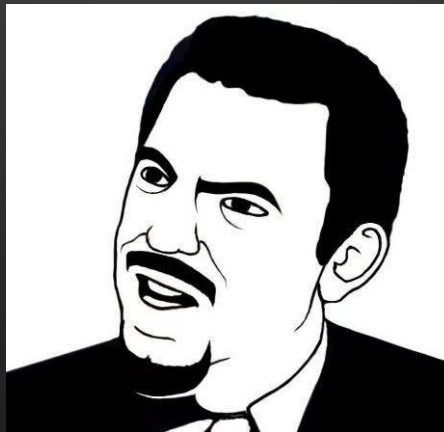


# 2LO Overview

Did you think it would be easy?



GData and “API Client” have different 2LO implementations!

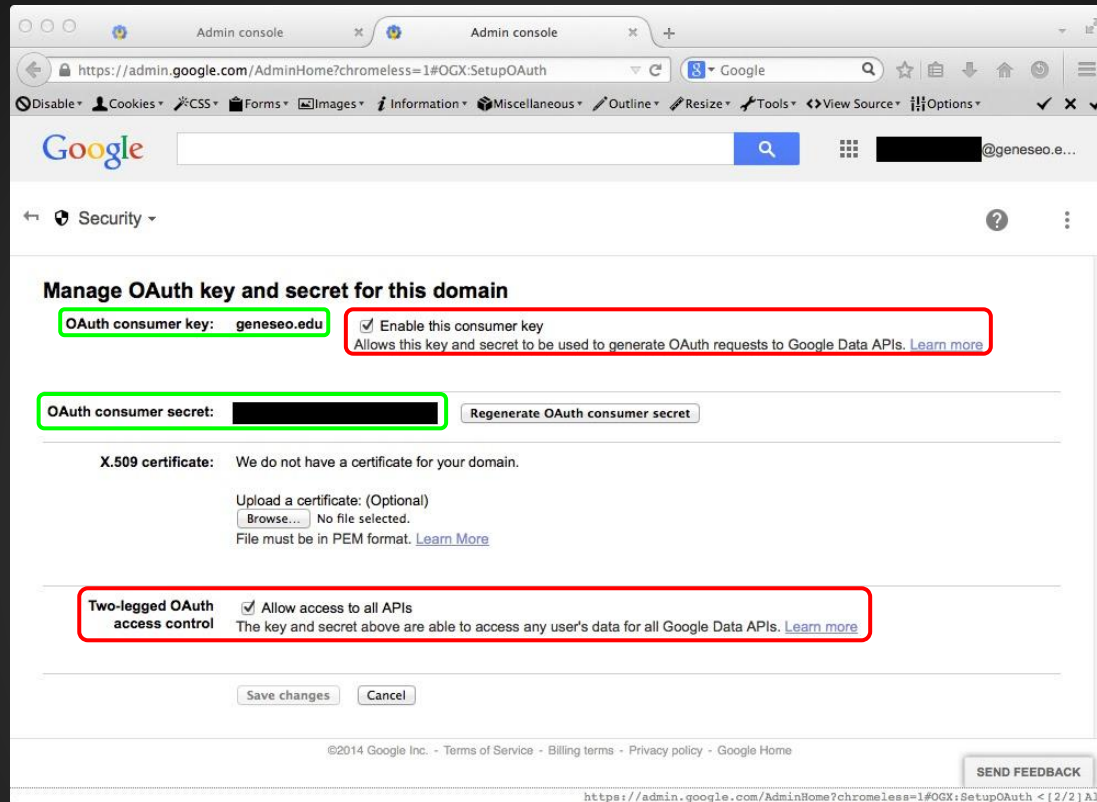




# GData 2LO Overview

1. Get Consumer Key and Consumer Secret
2. Ensure 2LO is enabled
3. Use Consumer Key and Secret to authenticate to GData API

# GData 2LO Steps 1 & 2



Get here via

1. Google Admin Console
2. Security
3. Advanced settings
4. Manage OAuth domain key

# GData 2LO Step 3 - Meat & Potatoes



```
import gdata.gauth
import gdata.contacts.client
```

```
SOURCE_APP_NAME = 'anything-you-want'
CONSUMER_KEY = 'your-consumer-key' # Generally, your GApps domain
CONSUMER_SECRET = 'your-consumer-secret'
requestor_id = 'gapps-user-email-address' # User you wish to access
```

```
two_legged_oauth_token = gdata.gauth.TwoLeggedOAuthHmacToken(
    CONSUMER_KEY, CONSUMER_SECRET, requestor_id)
```

```
contacts_client = gdata.contacts.client.ContactsClient
(source=SOURCE_APP_NAME)
```

```
contacts_client.auth_token = two_legged_oauth_token
```

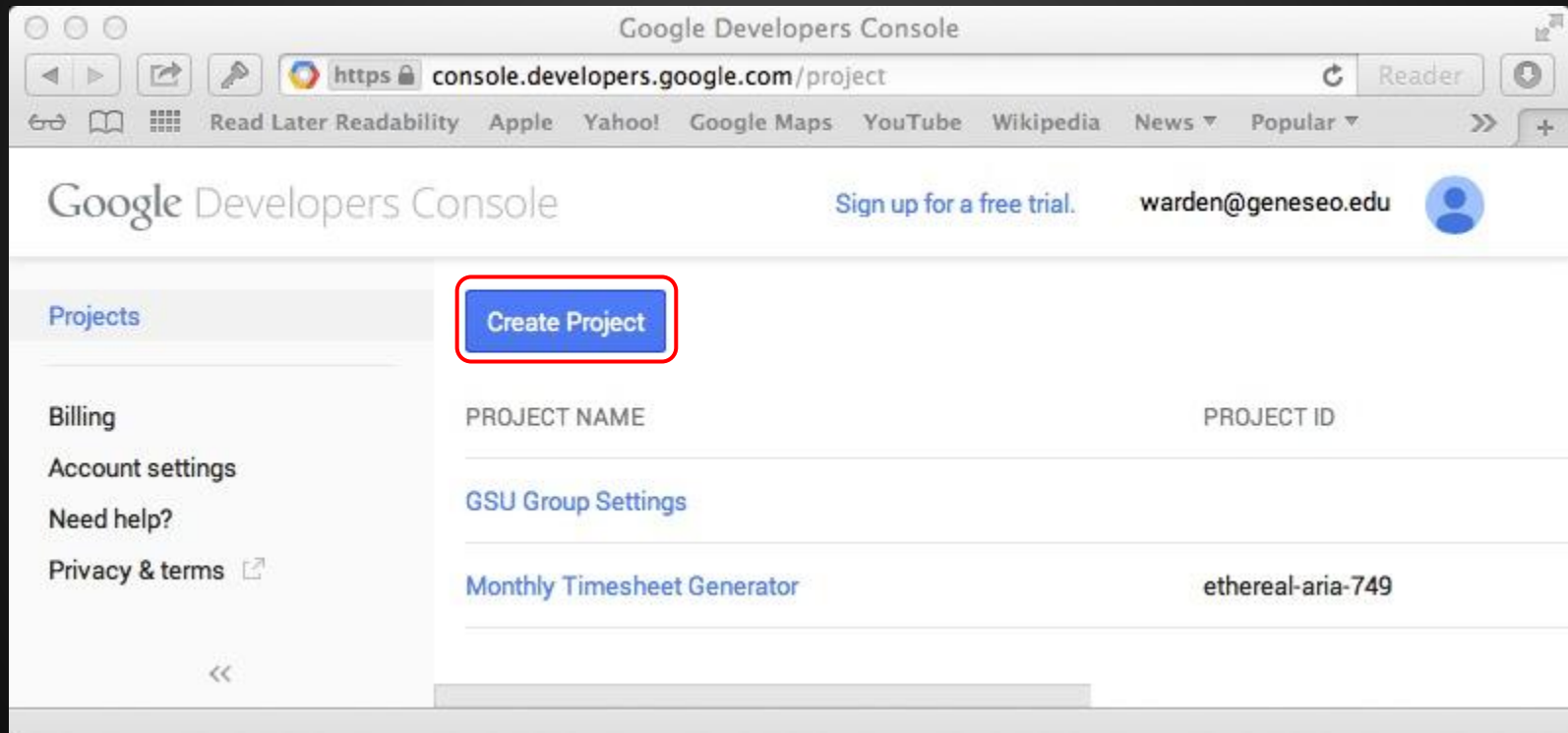
```
contacts_list = contacts_client.GetContacts()
for entry in contacts_list.entry:
    print entry.title.text
```

# “API Client” 2LO Overview

1. Create new project on Google Developers Console (GDC)
2. Enable APIs in GDC
3. Create service account in GDC
4. Enable scopes for service account in Google Admin Console
5. Use service account .p12 key to authenticate to “API Client” API

# “API Client” 2LO Step 1: Create Project

<https://console.developers.google.com>



Privileged account not required!

# “API Client” 2LO Step 2: Enable API(s)

The screenshot shows the Google Developers Console interface. On the left sidebar, under the 'APIs & auth' section, the 'APIs' link is highlighted with a red box and a red number '1'. In the main content area, the 'Enabled APIs' section lists several APIs that are already turned on. Below this, the 'Browse APIs' section has a search bar containing the word 'admin', which is also highlighted with a red box and a red number '2'. The search results show a table of APIs. The first row, 'Admin SDK', has its status set to 'OFF', which is highlighted with a red box and a red number '3'. The other APIs listed are 'Audit API', 'Groups Migration API', and 'Groups Settings API', all of which are also set to 'OFF'.

NAME	QUOTA	STATUS
BigQuery API	0%	ON
Google Cloud SQL		ON
Google Cloud Storage		ON
Google Cloud Storage JSON API		ON

NAME	QUOTA	STATUS
Admin SDK	Admin SDK lets administrators of enterprise domains to view and manage resources like user, 150,000 requests/day	OFF
Audit API	The Audit API allows domain administrators to view actions of users in their domain in various 10,000 requests/day	OFF
Groups Migration API	The Groups Migration API allows domain administrators to archive emails into Google groups. 500,000 requests/day	OFF
Groups Settings API	The Groups Settings API allows domain administrators to view and manage access levels and 100,000 requests/day	OFF

There are other interesting APIs besides Admin SDK!



# “API Client” 2LO Step 3: Create Service Account

The screenshot shows the Google Developers Console interface. The left sidebar has a 'Credentials' link highlighted with a red box and the number 1. The main content area shows the 'OAuth' section with a 'Create new Client ID' button highlighted with a red box and the number 2. A modal dialog titled 'Create Client ID' is open, showing the 'APPLICATION TYPE' section with 'Service account' selected (highlighted with a red box and the number 3). The 'Create Client ID' button in this dialog is highlighted with a red box and the number 4. Below the dialog, a 'New Public/Private key pair generated' message is shown, with the 'Okay, got it' button highlighted with a red box and the number 5.

Google Developers Console

Sign up for a free trial. warden@geneseo.edu

Projects

Wizards 2014

APIs & auth

1 Credentials

Consent screen

Push

Monitoring

Source Code

Compute

OAuth

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private.

Learn more

2 Create new Client ID

Create Client ID

APPLICATION TYPE

Web application  
Accessed by web browsers over a network.

3 Service account  
Calls Google APIs on behalf of your application instead of an end-user. Learn more

Installed application  
Runs on a desktop computer or handheld device (like Android or iPhone).

4 Create Client ID Cancel

New Public/Private key pair generated

The private key has been downloaded to your machine and serves as the only copy of this key.  
You are responsible for storing it securely.

Your private key's password is displayed below. it will not be shown again.

notasecret

You must present this password in order to use the private key.

5 Okay, got it

# “API Client” 2LO Step 4: Enable Scopes

Google Admin Console > Security > Advanced settings > Manage API client access

Google Developers Console

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private.

Service Account

CLIENT ID: 688062761320-sc02urs1cuv5qdankor09neklevtb4h6.apps.googleusercontent.com

EMAIL ADDRESS: 688062761320-sc02urs1cuv5qdankor09neklevtb4h6@developer.gserviceaccount.com

Authorized API clients

The following API client domains are registered with Google and authorized to access data for your users.

Client Name: Example: www.example.com

One or More API Scopes: Example: http://www.google.com/calendar/feeds/ (comma-delimited) [Authorize]

Learn more about registering new API clients

Scopes for users and user aliases	Meaning
<a href="https://www.googleapis.com/auth/admin.directory.user">https://www.googleapis.com/auth/admin.directory.user</a>	For all user and user alias operations, use this global user scope
<a href="https://www.googleapis.com/auth/admin.directory.user.readonly">https://www.googleapis.com/auth/admin.directory.user.readonly</a>	Use to limit the administrator's scope for retrieving users or user aliases

SEND FEEDBACK

“Authorize requests” in API documentation usually has available scopes

# “API Client” 2LO Step 5: Return of the Meat & Potatoes



```
import httplib2
import oauth2client.client
from apiclient.discovery import build

f = file('path-to-key-file', 'rb')
serviceacct_key = f.read()
f.close()

credentials = oauth2client.client.SignedJwtAssertionCredentials(
    'service-account-email-address', serviceacct_key,
    scope = [ 'scope-previously-authorized' ],
    sub = 'gapps-privileged-user-email-address' )

http = credentials.authorize(httplib2.Http())
directory = build('admin', 'directory_v1', http=http)

print directory.users().get(userKey='gapps-user-email-address').execute()
```

# GData and “API Client” Summary

## Client Libraries Bookmarks:

- GData: <https://developers.google.com/gdata/docs/client-libraries>
- “API Client”: <https://developers.google.com/discovery/libraries>

## For Python:

- GData: <http://gdata-python-client.googlecode.com/hg/pydocs/>
  - Monolithic documentation
- “API Client”: <https://developers.google.com/api-client-library/python/apis/>
  - Per-API documentation. Click the version link on an API in the link above, then “PyDoc reference for the X API”

**That's nice.**

How about something useful?

# How about pushed Google Apps logs?

<https://developers.google.com/admin-sdk/reports/v1/guides/push>



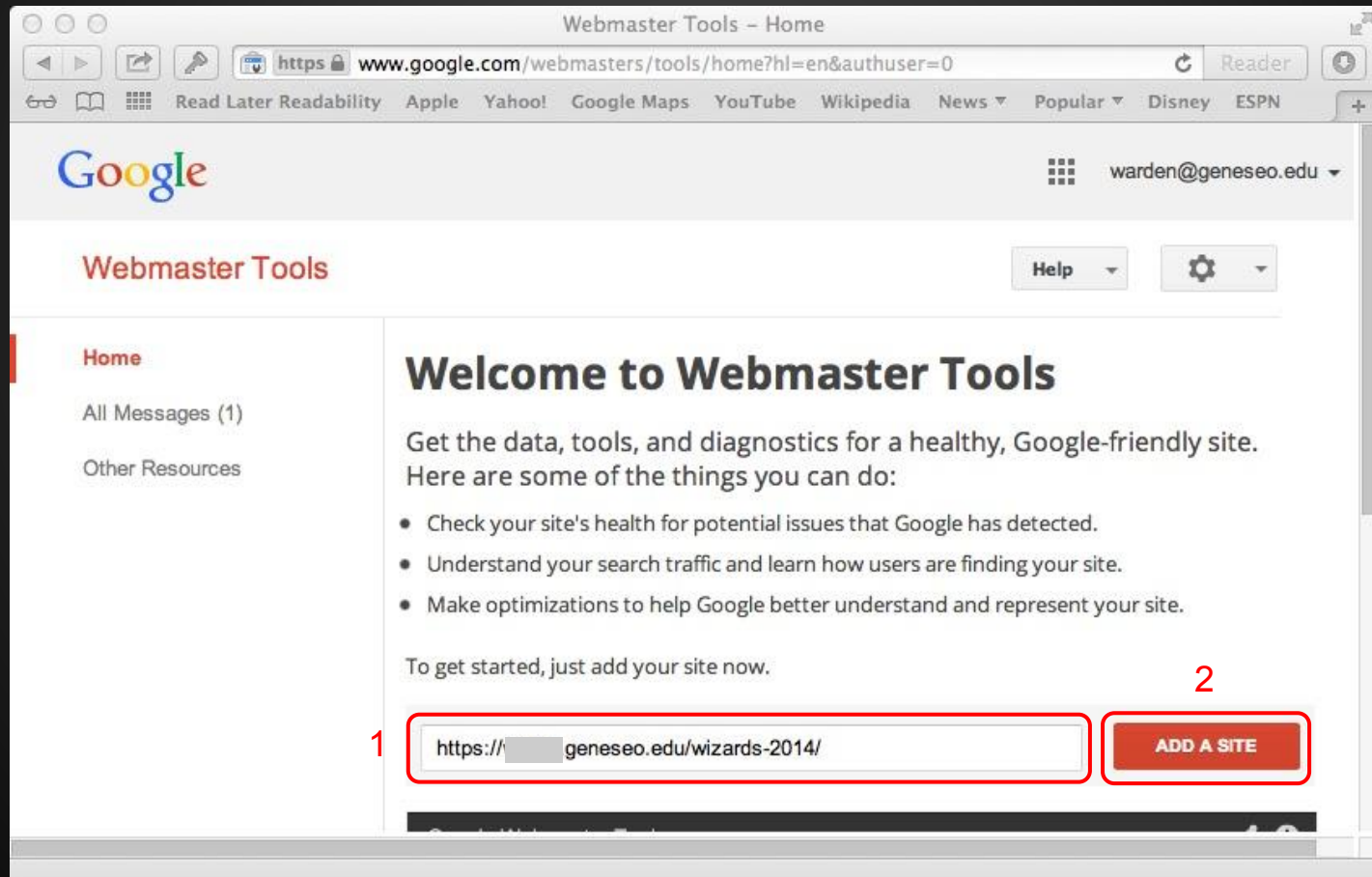
# Push Logs Overview

0. Follow “API Client” 2LO steps for both scopes of the Reports API in Admin SDK.
1. Verify ownership of a URL in Google Webmaster
2. Enable Push in Google Developers Console (GDC)
3. Create HTTP endpoint under URL
4. Initiate log push channel(s)
5. PROFIT



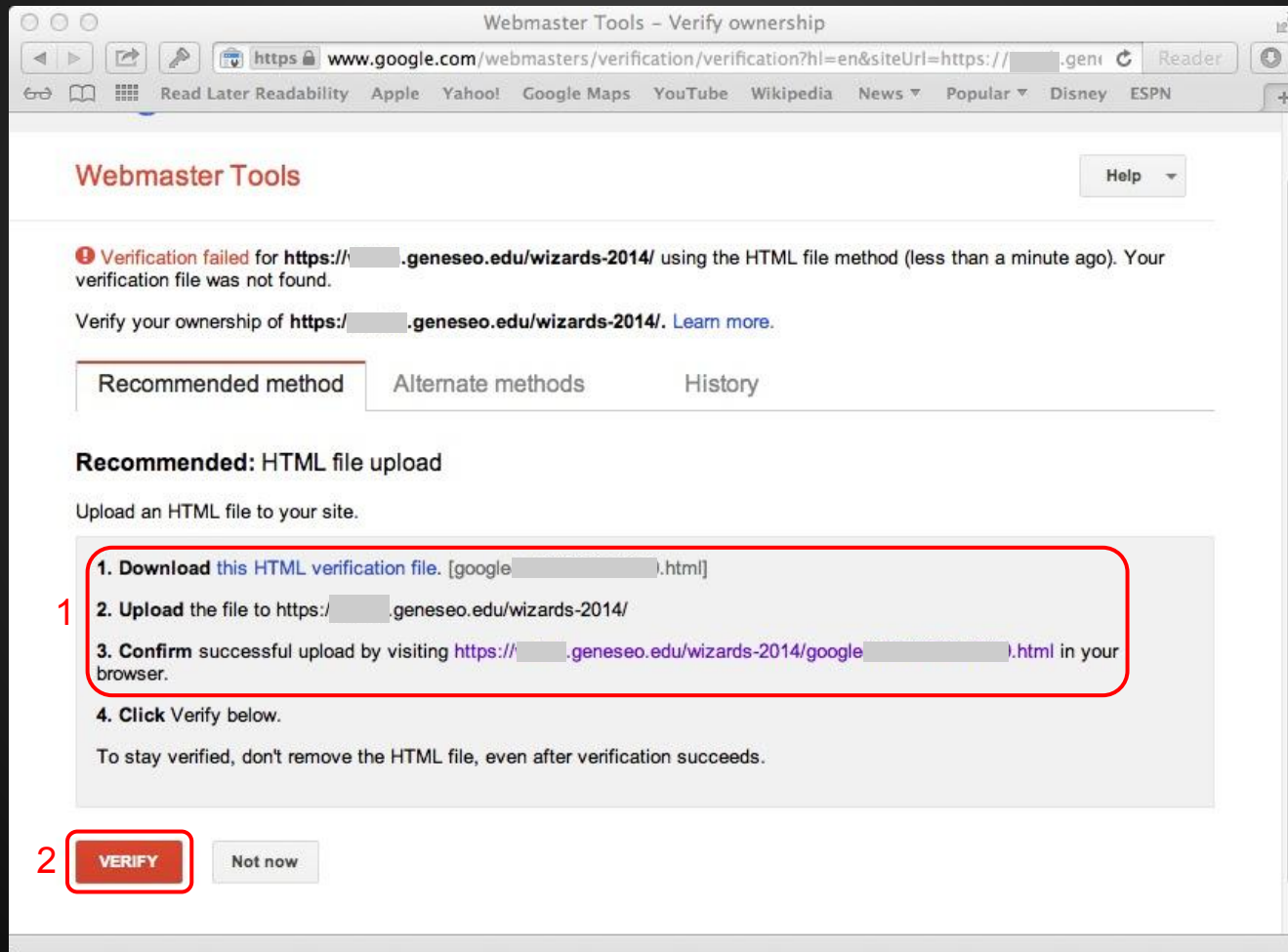
# Push Logs Step 1a: Add a URL in Webmaster Tools

<https://www.google.com/webmasters/tools/>

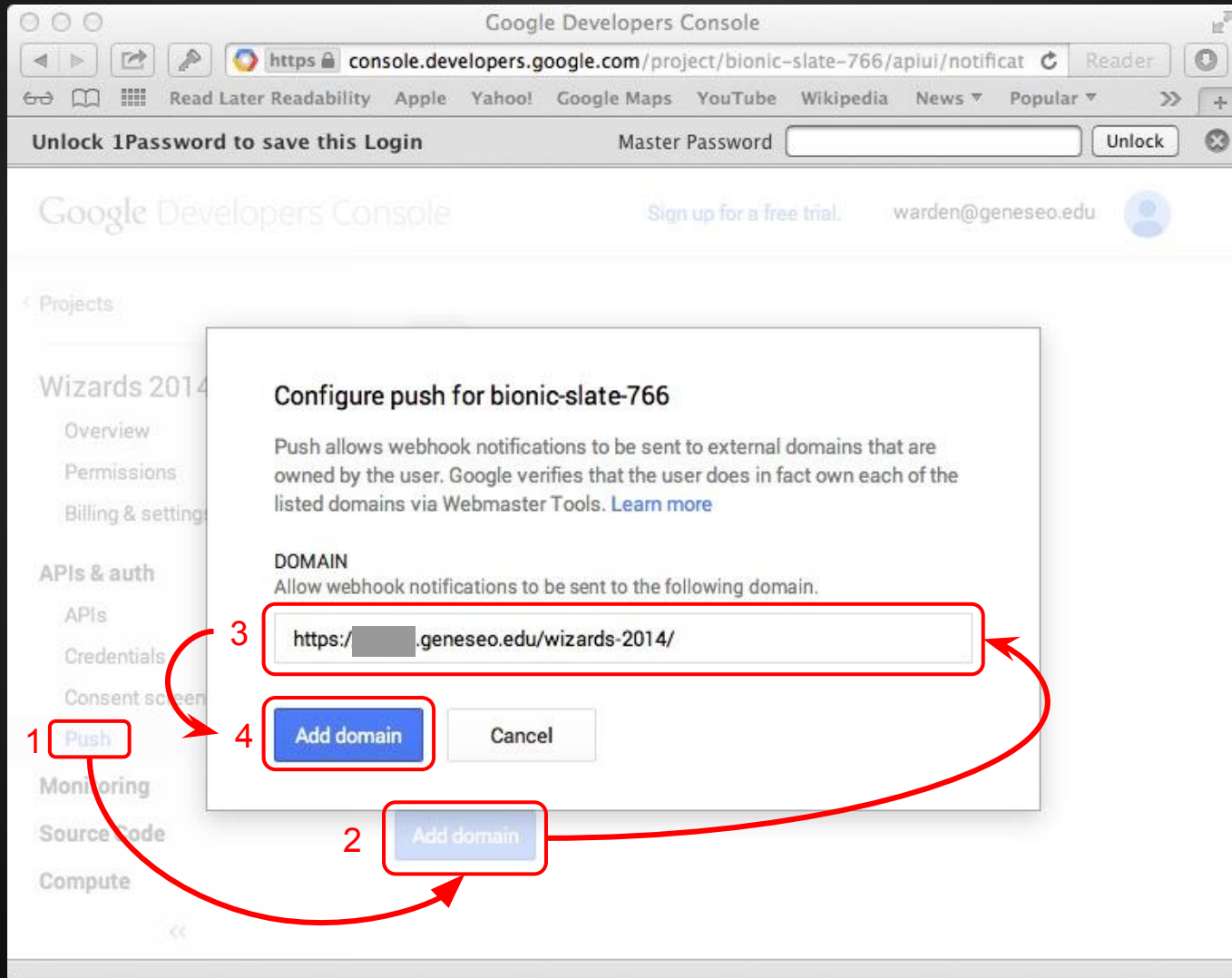


# Push Logs Step 1b: Verify a URL in Webmaster Tools

<https://www.google.com/webmasters/tools/>



# Push Logs Step 2: Enable Push for URL



# Push Logs Step 3:

## Script to dump HTTP requests to JSON

```
$log_file = 'path-to-log-file';
```

```
$headers = apache_request_headers();
```

```
$body = file_get_contents('php://input');
```

```
$payload = json_decode($body, $assoc=true);
```

```
if (json_last_error() != JSON_ERROR_NONE)  
    exit;
```

```
# Headers and payload have been successfully loaded in associative arrays.
```

```
# Merge, append newline and append to log file.
```

```
if (is_array($payload))
```

```
    $event = array_merge($headers, $payload);
```

```
else
```

```
    $event = $headers;
```

```
$event_string = json_encode($event) . "\n";
```

```
file_put_contents($log_file, $event_string, FILE_APPEND);
```

The PHP logo, consisting of the lowercase letters 'php' in a white, italicized, sans-serif font, centered within a solid blue rectangular background.

# Push Logs Step 4: Init Push Channels Meat & Potatoes



```
# Not shown: "API Client 2LO" setup of http variable
reports = discovery.build('admin', 'reports_v1', http=http)
import time
from datetime import datetime, timedelta
expire = datetime.now() + timedelta(hours=6) # Max allowed by google
expire_unix = int(time.mktime(expire.timetuple()))
expire_milliseconds = expire_unix * 1000
reports.activities().watch(userKey='all', applicationName='admin', body=dict(
    type='web_hook',
    address='https://example.geneseo.edu/wizards-2014/pushtolog.php',
    id='gsu-admin-{0}'.format(expire_unix), expiration=expire_milliseconds)).execute()
reports.activities().watch(userKey='all', applicationName='login', body=dict(
    type='web_hook',
    address='https://example.geneseo.edu/wizards-2014/pushtolog.php', id='gsu-login-
{0}'.format(expire_unix))).execute()
reports.activities().watch(userKey='all', applicationName='drive', body=dict(
    type='web_hook',
    address='https://example.geneseo.edu/wizards-2014/pushtolog.php', id='gsu-drive-
{0}'.format(expire_unix))).execute()
```



# PROFIT

Sometimes Google Apps Password Sync fails.  
Google password changes as they happen:

The screenshot shows the Kibana 3 Logstash Search interface. The browser address bar displays `kibana/index.html#/dashboard/file/logstash.json`. The search query is `type:googleapps AND X-Goog-Resource-State:"CHANGE_PASSWORD"`. The left sidebar shows the 'ALL EVENTS' section with a list of fields for filtering, including `@timestamp`, `@version`, `_id`, `_index`, `_type`, `Accept`, `Accept-Encoding`, `actor.callerType`, `actor.email`, `actor.profileId`, and `Connection`. The main panel displays a list of events, each with a JSON structure. A red box highlights a specific event's parameters, showing a list of email addresses, with the last one being `@geneseo.edu`.

Kibana 3 – Logstash Search

<https://kibana/index.html#/dashboard/file/logstash.json>

Logstash Search

a day ago to a few seconds ago

QUERY

`type:googleapps AND X-Goog-Resource-State:"CHANGE_PASSWORD"`

FILTERING

GRAPH

ALL EVENTS

Fields

All (852) / Current (33)

Type to filter...

- ☐ @timestamp
- ☐ @version
- ☐ \_id
- ☐ \_index
- ☐ \_type
- ☐ Accept
- ☐ Accept-Encoding
- ☐ actor.callerType
- ☐ actor.email
- ☐ actor.profileId
- ☐ Connection

events

0 to 60 of 60 available for paging

```
{
  "type": "USER_SETTINGS",
  "name": "CHANGE_PASSWORD",
  "parameters": {
    "name": "USER_EMAIL",
    "value": "@geneseo.edu"
  }
}
```

# It's Not All Rainbows and Unicorns



- You need to renew those Push channels every 6 hours, which sometimes fails
- Logstash has problems fully parsing some of the deep JSON fields

# One last thing...

<https://github.com/dfwarden/SUNYWizards2014-GoogleAppsAPIs>



Pull requests welcome!