Content Migration with Python

Using tools such as Google Sheets, Python 3, WordPress REST JSON API

Brent Deverman / May 9, 2020

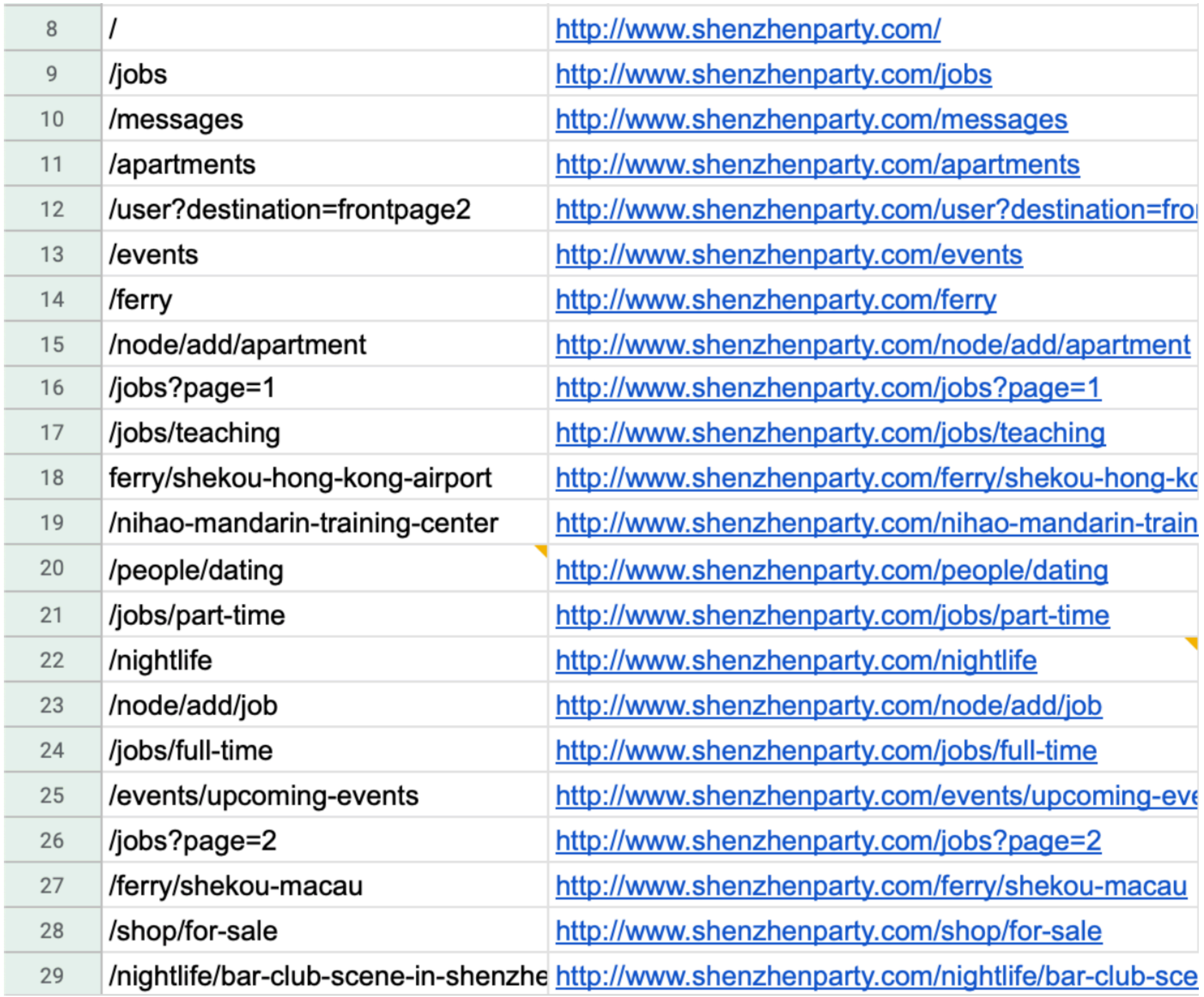
Brent Deverman

Product Manger and Entrepreneur

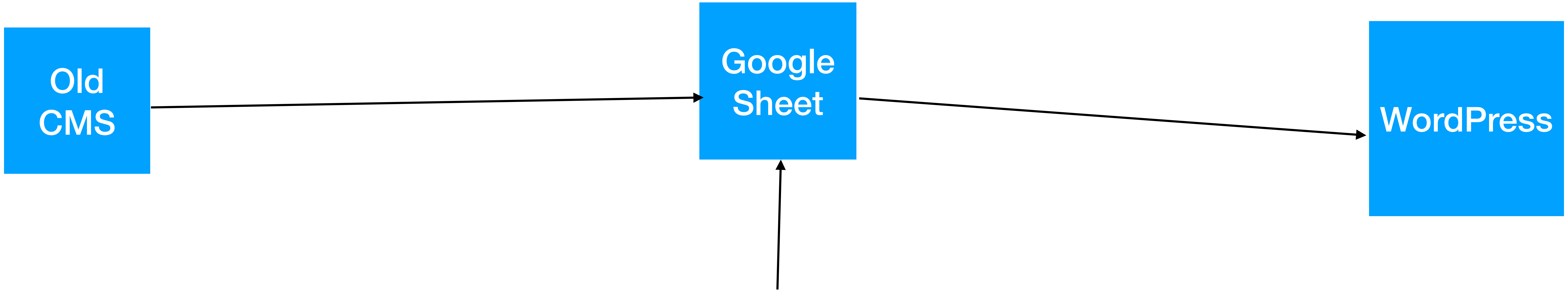
- Previously worked for DFS, HSBC & CNN
- Python scripts are for my business ShenzhenParty.com
- Contact: brent@deverman.org
- Twitter: @bdeverman
- LinkedIn: Brent Deverman

Migration Goal

- Copy most popular posts from old CMS to WordPress while preserving SEO and tracking progress in Google Sheets
- Start with an export of top posts in Google Sheets
- Match content URLs with Original CMS info including post IDs
- Use the spreadsheet to as a control panel to map URLs and track migration progress



8	/	http://www.shenzhenparty.com/
9	/jobs	http://www.shenzhenparty.com/jobs
10	/messages	http://www.shenzhenparty.com/messages
11	/apartments	http://www.shenzhenparty.com/apartments
12	/user?destination=frontpage2	http://www.shenzhenparty.com/user?destination=fro
13	/events	http://www.shenzhenparty.com/events
14	/ferry	http://www.shenzhenparty.com/ferry
15	/node/add/apartment	http://www.shenzhenparty.com/node/add/apartment
16	/jobs?page=1	http://www.shenzhenparty.com/jobs?page=1
17	/jobs/teaching	http://www.shenzhenparty.com/jobs/teaching
18	ferry/shekou-hong-kong-airport	http://www.shenzhenparty.com/ferry/shekou-hong-kc
19	/nihao-mandarin-training-center	http://www.shenzhenparty.com/nihao-mandarin-train
20	/people/dating	http://www.shenzhenparty.com/people/dating
21	/jobs/part-time	http://www.shenzhenparty.com/jobs/part-time
22	/nightlife	http://www.shenzhenparty.com/nightlife
23	/node/add/job	http://www.shenzhenparty.com/node/add/job
24	/jobs/full-time	http://www.shenzhenparty.com/jobs/full-time
25	/events/upcoming-events	http://www.shenzhenparty.com/events/upcoming-eve
26	/jobs?page=2	http://www.shenzhenparty.com/jobs?page=2
27	/ferry/shekou-macau	http://www.shenzhenparty.com/ferry/shekou-macau
28	/shop/for-sale	http://www.shenzhenparty.com/shop/for-sale
29	/nightlife/bar-club-scene-in-shenzhe	http://www.shenzhenparty.com/nightlife/bar-club-sce



Old CMS

Google Sheet

WordPress

5000

Pages

Python Libraries

Here are the solutions I used:

```
# Call the REST API
import json # convert json API call to dict
import requests # make json API call

# Talk to Google Sheets
import gspread
from oauth2client.service_account import ServiceAccountCredentials

# Utility Libraries for string manipulation
from datetime import datetime # Convert strings to dates
import csv # Export/Import data to CSV files
import pprint # Helpful for debugging
import base64 # Need to encode for headers to create posts with
Application Passwords
```

WordPress Plugins

- REST API (Built-in)
    - In json format for reading and writing
- Application Passwords
    - For creating WP posts

Retrieve Data About Published Posts from Old CMS

Two step process for my CMS; Example code for WordPress JSON API

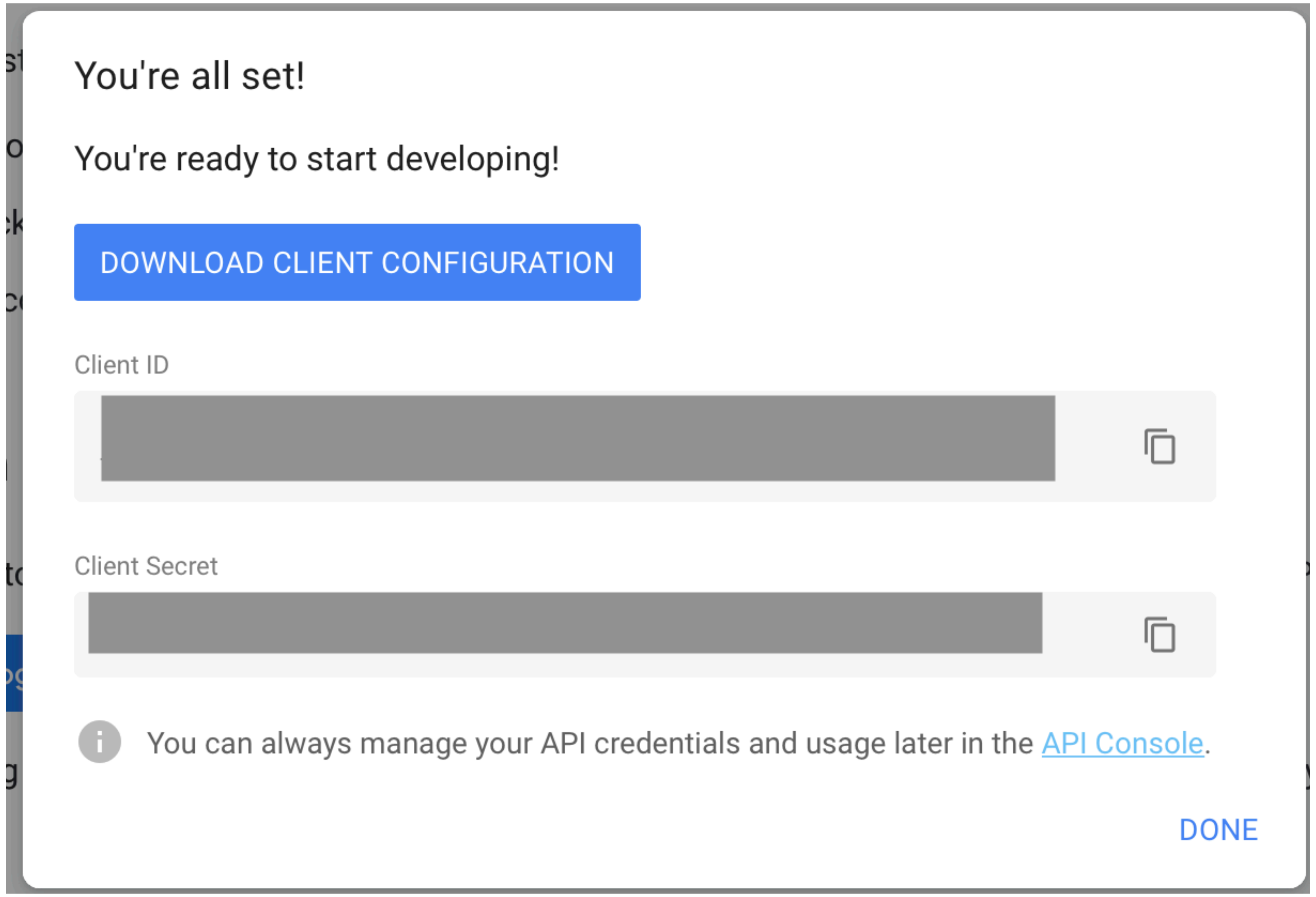
- Loop list of all posts on the website
- Loop again to get details of post
    - Key piece of information the URL path (URI)
    - Filter out specific content that should not be migrate

```python
#!/usr/bin/env python3

import json # convert json file to dict
import requests # get web url information
import csv

lastpage = False
count = 1
line = 1

with open("nodes.csv", mode='w', buffering=-1) as nodefile: # write to a file
    fieldnames = ['#', 'id', 'type', "title", "date", "link"]
    writer = csv.DictWriter(nodefile, fieldnames=fieldnames)

    writer.writeheader()

    while lastpage == False:
      url = f'https://www.example.com/wp-json/wp/v2/posts/?page={count}'
      j = requests.get(url).content
      d = json.loads(j)
      if "code" in d:
        lastpage = True
      count += 1
      for x in d:
        writer.writerow({'#': line, 'id': x["id"], 'type': x["type"], 'title': x["title"]["rendered"], 'date': x["date"], 'link': x["link"]})
        line += 1
```

Google Sheets Access Setup

Setup REST API

- Reviewed multiple Google Sheet python libraries and GSpread seemed to be the easiest.
- Enable the Google Sheets API
- Download the credentials.json in the same directory as your python script



You're all set!

You're ready to start developing!

Client ID

[redacted]

Client Secret

[redacted]

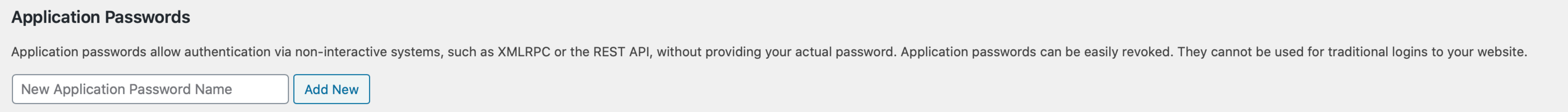
You can always manage your API credentials and usage later in the API Console.

Match Data & Update Google Sheets

```
#!/usr/bin/env python3
# Utility function for list lookup so we don't throw error
def index(a_list, value):
    try:
        return a_list.index(value)
    except ValueError:
        return None

import gspread # manipulate popular gsheet
import pprint # for debugging
from oauth2client.service_account import ServiceAccountCredentials #
Google sheet access
import json  # convert json file to dict
from datetime import datetime
import csv

scope = ['https://spreadsheets.google.com/feeds',
         'https://www.googleapis.com/auth/drive']

# Get credentials
credentials = ServiceAccountCredentials.from_json_keyfile_name('/
path/to/credentials.json', scope)

# Authenticate with API for Google Sheet
client = gspread.authorize(credentials)

sp = client.open('My Worksheet') # Grab the main worksheet we will
be using
work_sheet = sp.sheet1 # Get actual Sheet

# Grab the worksheet of content we propose to not migrate because we
didn't find it in the top 5K
```

```
reject_sheet = sp.worksheet("Propose Delete")

# Get URL fragements to see if the content exists
urlfrag = work_sheet.col_values(1)

combined = [] # Store pages that exist in the top 5K here
rejected = [] # Store pages that don't exist in the top 5K here

# sift through urls that already exist in the google sheet
with open("content.csv", mode='r', buffering=-1) as contentfile:

   # Open old pages file for reading as dictionary
   nodereader = csv.DictReader(contentfile)

   # Loop through contents of file
   for row in nodereader:

      idx = index( urlfrag, "/" + row["path"] )
      if idx is not None:
         row["sheetrow"] = idx+1
         combined.append(row)
      else:
         row["sheetrow"] = idx
         rejected.append(row)

# Example of updating the Google Sheet
reject_sheet.update("A1", "Test Me")
```

Create Post with an Application Password



Application Passwords

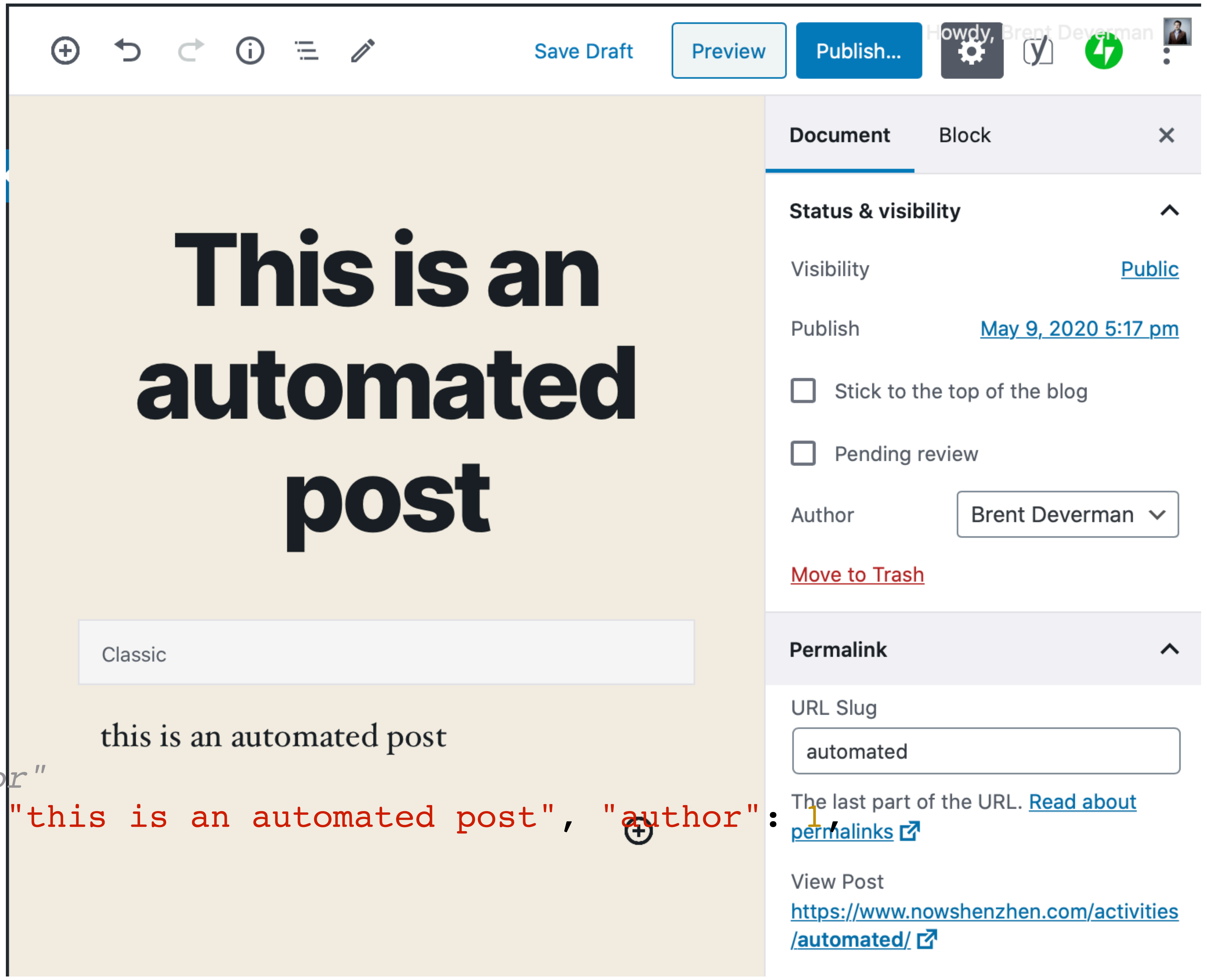
Application passwords allow authentication via non-interactive systems, such as XMLRPC or the REST API, without providing your actual password. Application passwords can be easily revoked. They cannot be used for traditional logins to your website.

- You will need the account WordPress username
- Then Application Password associated with that username this can be created on the profile page after Activating the Application Password Plugin

Create Posts in WordPress

Sample to code to send a post

```
#!/usr/bin/env python3

import json # convert json file to dict
import requests # get web url information
import base64 # need to encode for headers

# Post content, you have to find out the ID of your WordPress account for "author"
postitem = {"status": "draft", "title": "This is an automated post", "content": "this is an automated post", "author": 1,
"excerpt": "test", "format": "standard", "slug": "automated"}
# Applicaiton Passwords plugin
apppass = "INSERTAPPPASSWORDHERE"
username = "admin" # Change to your username here
# build authentication header and encode it as base64 then convert to string
authhead = username + ":" + apppass
authhead = base64.urlsafe_b64encode(authhead.encode("utf-8"))
authhead = str(authhead, "utf-8") # Ensure correct encoding

# 'Authorization': 'Bearer '+user.token
headers = {"Authorization": 'Basic '+ authhead}
url = f'https://www.example.com/wp-json/wp/v2/posts'

# return content of POST request
j = requests.post(url, json=postitem, headers=headers).content

# can store new post information such as ID in Google Sheet
```



This is an automated post

Classic

this is an automated post

Status & visibility

Visibility: Public

Publish: May 9, 2020 5:17 pm

□ Stick to the top of the blog

□ Pending review

Author: Brent Deverman

Move to Trash

Permalink

URL Slug: automated

The last part of the URL. Read about permalinks

View Post
https://www.nowshenzhen.com/activities/automated/

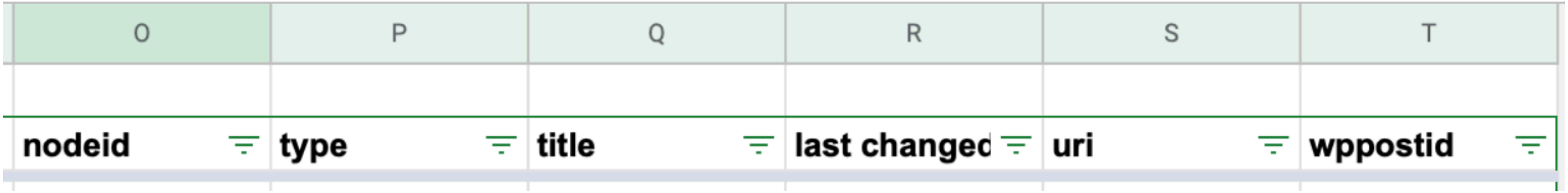
Search WordPress Posts by URI

Bonus Tip

- WordPress API allows to query for a post by slug
    - https://www.example.com/wp-json/wp/v2/posts?slug=post-slug
- (Works best if post is published.)

Pulling it All Together

Using the Google Sheet as the control panel



O	P	Q	R	S	T
nodeid	type	title	last changed	uri	wppostid

- Bring Old CMS post IDs and new WordPress Post IDs into Google Sheets
- Allows Python script to more easily migrate content by post ID
- Can Run the migration multiple times until you get right
- Ability to update the published status in the spreadsheet of the new posts

Thank You

Email me: brent@deverman.org