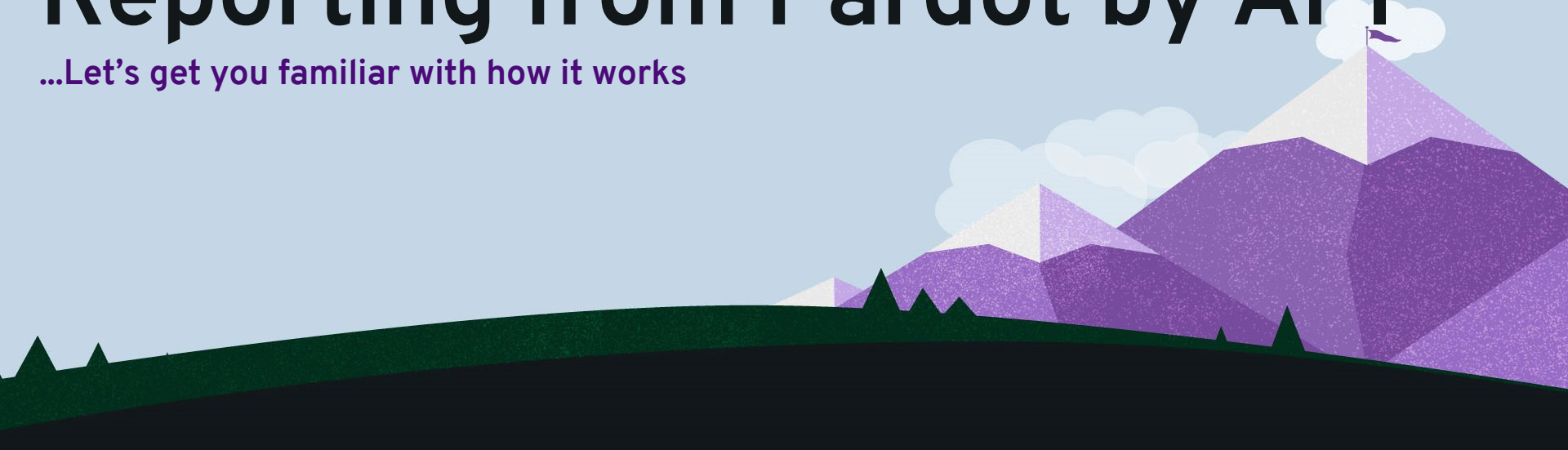


Data Science & Ad Hoc Reporting from Pardot by API

...Let's get you familiar with how it works



Data Science & Ad Hoc Reporting from Pardot by API

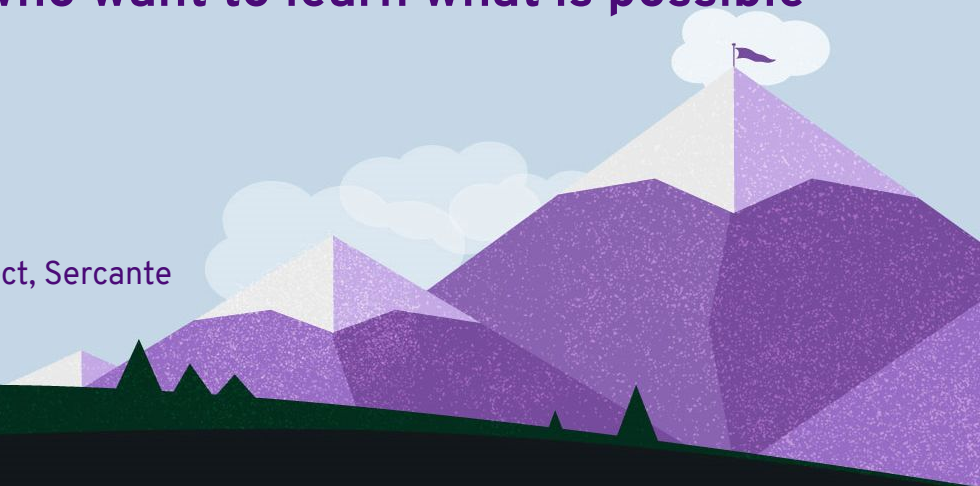
For programmers or business users who want to learn what is possible

Advanced | Tech



Mike Creuzer

CRM & Marketing Automation Solution Architect, Sercante



What is Data Science?

~~A data scientist is someone who creates programming code, and combines it with statistical knowledge to create insights on business data.~~

Kinda sounds like Marketing Reporting???

Uh-oh

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from noisy, structured and unstructured data, and apply knowledge and actionable insights from data across a broad range of application domains. Data science is related to data mining, machine learning and big data.

https://en.wikipedia.org/wiki/Data_science

Uh what???

Never mind all that... we get cool tools...

Introducing Jupyter Notebook

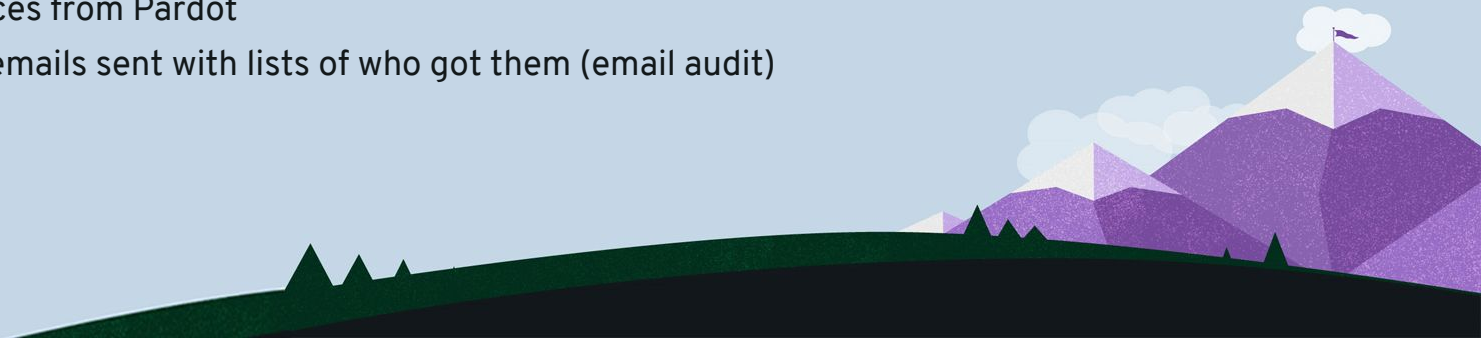
The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.



Ad Hoc Reporting with Pardot API

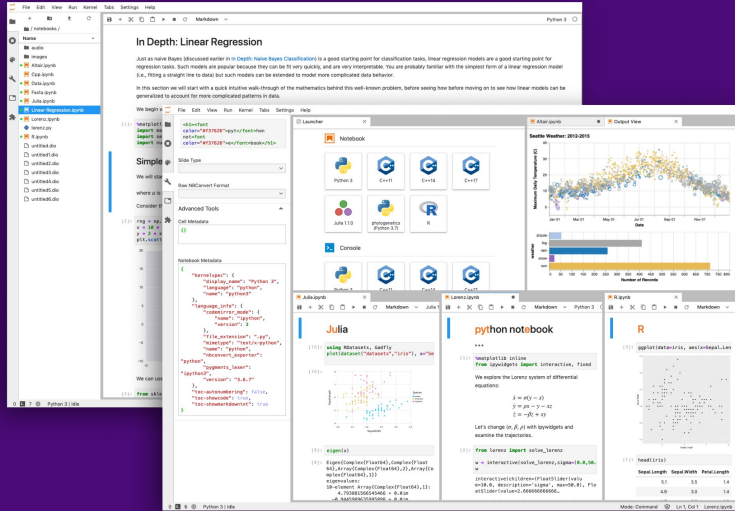
What to expect in this session:

- Intro to Jupyter Notebook
- Intro to Pardot API
- Get CRM IDs of deleted prospects for further analysis
- Export email bounces from Pardot
- Create folders for emails sent with lists of who got them (email audit)



Intro to Jupyter Notebook

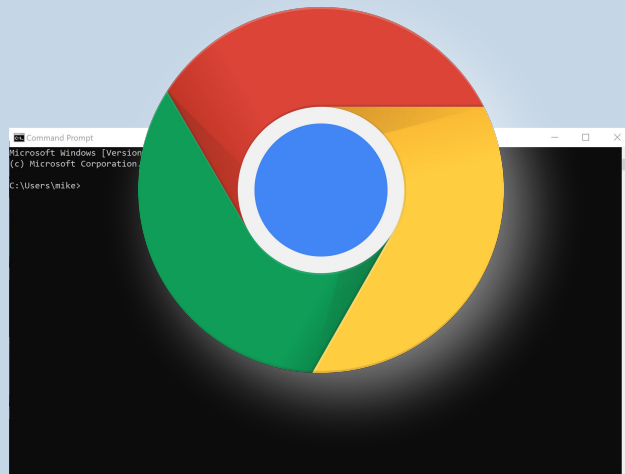
A lot of programming, a bit of excel, a bit of reporting



Web Based UI

But served locally

- Run a command line based 'server' (Kernel)
- Open URL in web browser
- Do work
- Profit



Installation

Windows/Mac/Linux

- Command line installation
- Docker Installation
- VSCode integration
- Desktop App Installation

<https://jupyter.org/install>

<https://jupyter-docker-stacks.readthedocs.io/en/latest/>

<https://code.visualstudio.com/docs/datascience/jupyter-notebooks>

<https://blog.jupyter.org/jupyterlab-desktop-app-now-available-b8b661b17e9a>

Quick Start

Non-obvious UI

For CLI installations

- Start tool run cli/start docker container/etc
- Find URL with security token
- Paste URL into web browser

For Desktop App

- Open Application

File Format

.ipynb

The safe file format has code, metadata, markdown, data all smushed together into a single file.

The Jupyter Notebook file format is JSON-based with extension . ipynb, after the project's predecessor, IPython. The format is open and exhaustively documented.

Beware Login details.... This would be baked into the file in cleartext too...



Intro to Pardot API

What is an API

Application Programming Interface

An application programming interface (API) is a connection between computers or between computer programs. It is a type of software interface, offering a service to other pieces of software.

<https://en.wikipedia.org/wiki/API>



What can the Pardot API do for us?

CRUD

Create / Read / Update / Delete

- Prospects
- Visitor Activities
- Custom Redirects
- Emails
- Lists
- more...

Versions 3/4/5

V3 Non AMPSEA enabled orgs

V4 AMPSEA enabled orgs

V5 New (in progress currently) which will replace V3/4 in time.

How to use the Pardot API

Code or something else?

Code

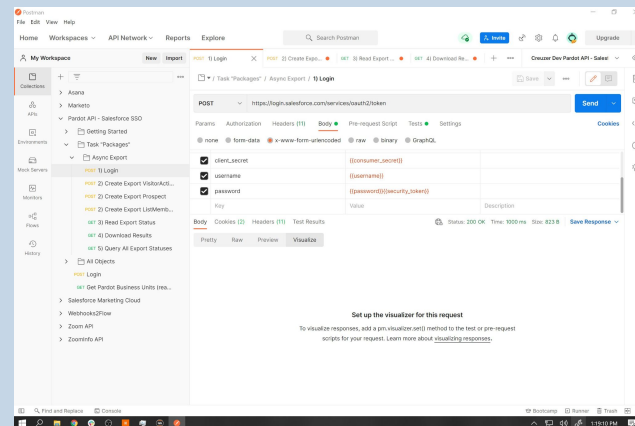
```
import requests
import json

url = 'https://api.pardot.com/v4/contacts/1'

headers = {
    'Content-Type': 'application/json',
    'Authorization': 'Bearer: ' + api_key,
    'Pardot-Business-Unit-ID': api_key
}

response = requests.get(url, headers=headers, data=body)
response_json = json.loads(response.text)
print(response_json)
```

Postman



Authentication

Salesforce SSO

- Salesforce user
 - Identity License
- Pardot User
 - Pardot Role controls access

Each integration ought to have it's own login

Production / dev / sandboxes will have different URLs

Alternatively

- JWT
- Certificate based



Examples

Demo Time!

Work with deleted prospects

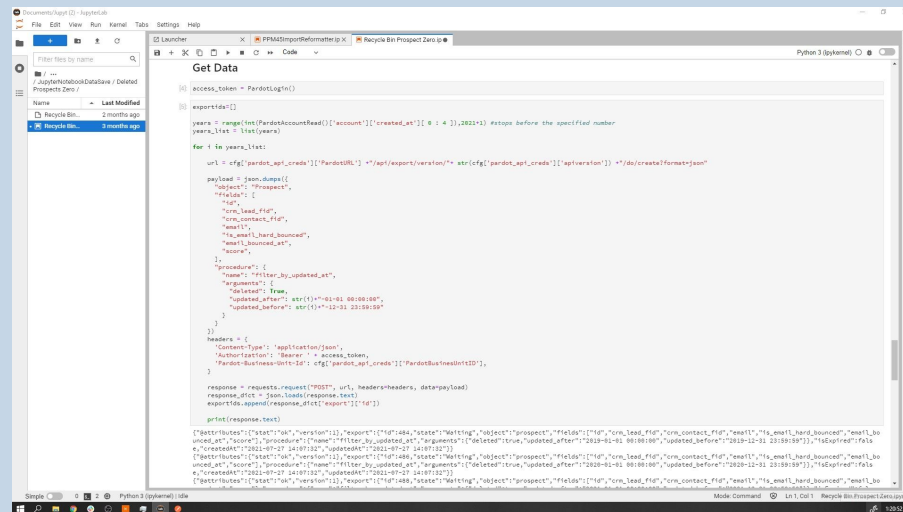
Get CRM IDs of deleted prospects for further analysis.

- Prospect Export
 - "deleted": True
- Save to .csv

One step further....

Deleted Prospects Zero

- Prospect Export
 - "deleted": True
- Update prospects with new score



```

Get Data
exportId()
access_token = PardotLogin()
exportId()

years = range(int(PardotAccountRead()['account']['created_at'][:4]), 2021+1) #stop before the specified number
years_list = list(years)

for i in years_list:
    url = f'api/pardot_apl_creds?{"&format=URL"}{"&format=JSON"}' + f'api/report/veration?{"&format=JSON"}' + f'api/create/format=json'

    payload = json.dumps({
        'subject': 'Prospect',
        'fields': [
            'id',
            'crm_lead_flag',
            'crm_contact_flag',
            'email',
            'email_hard_bounced',
            'email_bounced_at',
            'reason'
        ],
        'procedure': {
            'name': 'filter_by_updated_at',
            'arguments': {
                'deleted': True,
                'updated_after': str(i)+'-01-01 00:00:00',
                'updated_before': str(i)+'-12-31 23:59:59'
            }
        }
    })

    headers = {
        'Content-Type': 'application/json',
        'Authorization': 'Bearer ' + access_token,
        'Pardot-Business-Unit-ID': f'api/pardot_apl_creds?{"&format=URL"}{"&format=JSON"}'
    }

    response = requests.post(PDOS, url, headers=headers, data=payload)
    response_dict = json.loads(response.text)
    exportId.append(response_dict['exportId'])

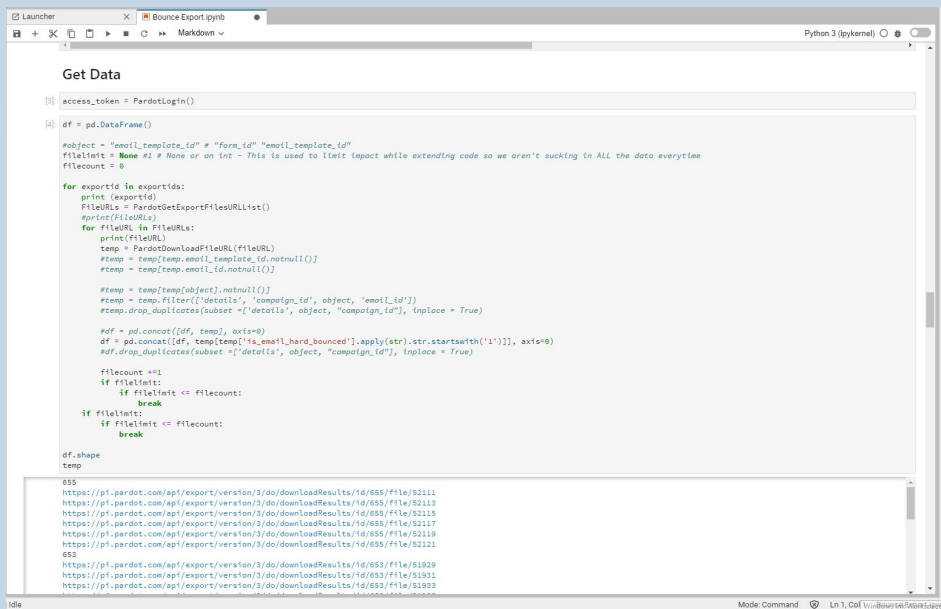
    print(response.text)

["{"id": "1234567890", "reason": "None", "updated_at": "2021-01-01 00:00:00", "deleted": true, "updated_after": "2021-01-01 00:00:00", "updated_before": "2021-12-31 23:59:59"}, {"id": "1234567890", "reason": "None", "updated_at": "2021-01-01 00:00:00", "deleted": true, "updated_after": "2021-01-01 00:00:00", "updated_before": "2021-12-31 23:59:59"}, {"id": "1234567890", "reason": "None", "updated_at": "2021-01-01 00:00:00", "deleted": true, "updated_after": "2021-01-01 00:00:00", "updated_before": "2021-12-31 23:59:59"}]
  
```

Export from Pardot

Export email bounces from Pardot

- Prospect Export
 - "is_email_hard_bounced",
 - "email_bounced_at",
 - "email_bounced_reason",
- Filter data to include only records that bounced
- Save to .csv



```

Get Data

access_token = PardotLogin()

df = pd.DataFrame()

object = "email_template_id" # "form_id" "email_template_id"
filelimit = None # None or an int - This is used to limit impact while extending code so we aren't sucking in ALL the data everytime
filecount = 0

for exportid in exportids:
    print(exportid)
    fileURLs = PardotGetExportFileURLList()
    #print(fileURLs)
    for fileURL in fileURLs:
        print(fileURL)
        temp = PardotDownloadFileURL(fileURL)
        #temp = temp[temp['email_template_id'].notnull()]
        #temp = temp[temp['email_id'].notnull()]

        #temp = temp[temp['object'].notnull()]
        #temp = temp.filter('details', 'campaign_id', object, 'email_id')
        #temp.drop_duplicates(subset = ['details', object, 'campaign_id'], inplace = True)

        #df = pd.concat([df, temp], axis=0)
        df = pd.concat([df, temp[temp['is_email_hard_bounced'].apply(str).str.startswith('1')]], axis=0)
        #df.drop_duplicates(subset = ['details', object, 'campaign_id'], inplace = True)

        filecount +=1
        if filelimit:
            if filelimit <= filecount:
                break
    if filelimit:
        if filelimit <= filecount:
            break

df.shape
temp

655
https://p1.pardot.com/api/export/ver/3/ds/downloadResults/4/655/f1e/52111
https://p1.pardot.com/api/export/ver/3/ds/downloadResults/4/655/f1e/52113
https://p1.pardot.com/api/export/ver/3/ds/downloadResults/4/655/f1e/52115
https://p1.pardot.com/api/export/ver/3/ds/downloadResults/4/655/f1e/52117
https://p1.pardot.com/api/export/ver/3/ds/downloadResults/4/655/f1e/52119
https://p1.pardot.com/api/export/ver/3/ds/downloadResults/4/655/f1e/52121
658
https://p1.pardot.com/api/export/ver/3/ds/downloadResults/4/653/f1e/51929
https://p1.pardot.com/api/export/ver/3/ds/downloadResults/4/653/f1e/51931
https://p1.pardot.com/api/export/ver/3/ds/downloadResults/4/652/f1e/51933
  
```

Create email audit

Create folders for emails sent with lists of who got them (email audit)

- Visitor Activity Export
- Filter for just emails sent
- Group and sort by email send
- Create folder for each email sent
- Get email HTML and TEXT versions and write to folder
- Get prospect details needed
- Save recipients of email to .csv

```

def label_type (row):
    if row['type'] == 8 :
        return 'Sent'
    if row['type'] == 12 :
        return 'Unsubscribe Page'
    if row['type'] == 13 :
        return 'Bounced'
    if row['type'] == 14 :
        return 'Spam Complaint'
    if row['type'] == 15 :
        return 'Email Preference Page'
    if row['type'] == 16 :
        return 'Reunsubscribed'
    return 'Other'

# get a list of email template IDs
emails=df['list_email_id'].unique().tolist()
emailtemplates=df[df['list_email_id']==emails]['email_template_id'].unique().tolist()
emailtemplates

# Taken from https://stackoverflow.com/a/908612/119527
def mkdir_p(path):
    try:
        os.makedirs(path)
    except OSError as exc: # Python >2.5
        if exc.errno == errno.EEXIST and os.path.isdir(path):
            pass
        else: raise

def safe_open_w(path):
    """ Open 'path' for writing, creating any parent directories as needed.
    """
    mkdir_p(os.path.dirname(path))
    return open(path, 'w', encoding='utf-8')

for email in emails:

```

Other Uses

- Preprocess .csv file for import
 - Repeating grindy excel steps with no extra work
- Merge multiple data into a single field
 - Create HTML and TEXT email snippets to import
- Dynamic Content & tag usage in email templates
 - Look at all email templates for PML/HML tags
- Link Check your Custom Redirects
 - Linkrot is the WORST!
- Export all files
 - Migrations anybody?
- Migrator (files, custom fields, Custom Redirects)
 - Not everything can be migrated, some stuff can!



Thanks for listening!



creuzerm



creuzerm

You can find me on the social media links
above or hit me up at creuzer@sercante.com

Thank You To Our Sponsors

TITLE SPONSORS



SERCANTE

PLATINUM SPONSORS



feedotter

SAASCEND



terminus



GetFeedback™



QUALIFIED

GOLD SPONSORS



THE SPOT
FOR PARDOT by SERCANTE

SILVER SPONSORS

FIONTA

CLOUDTEGIC