

# Play with Google APIs

Ratchanon Panmas  
@ratchanonp

```
...org = filterByOrg ? study.lead_organization == filterByOrg : true  
...status = filterByStatus ? study.status == filterByStatus : true  
...matchStatus) {
```

```
...function filterStudies({ studies, filterByOrg  
...studies.filter(study
```



# Agenda

What we gonna do today

What is API & Prepare	10 min
-----------------------	--------

---

Python - Request Library	20 min
--------------------------	--------

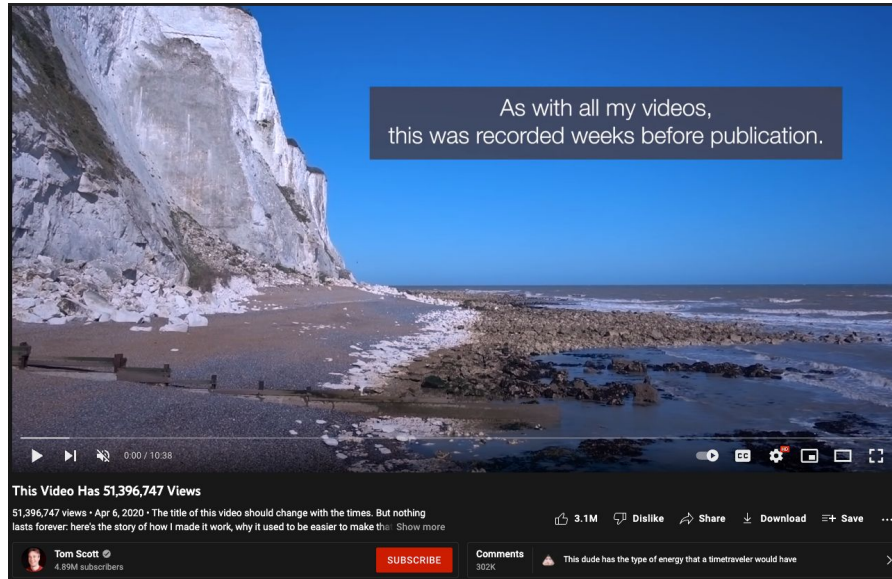
---

Google APIs Document Tour	15 min
---------------------------	--------

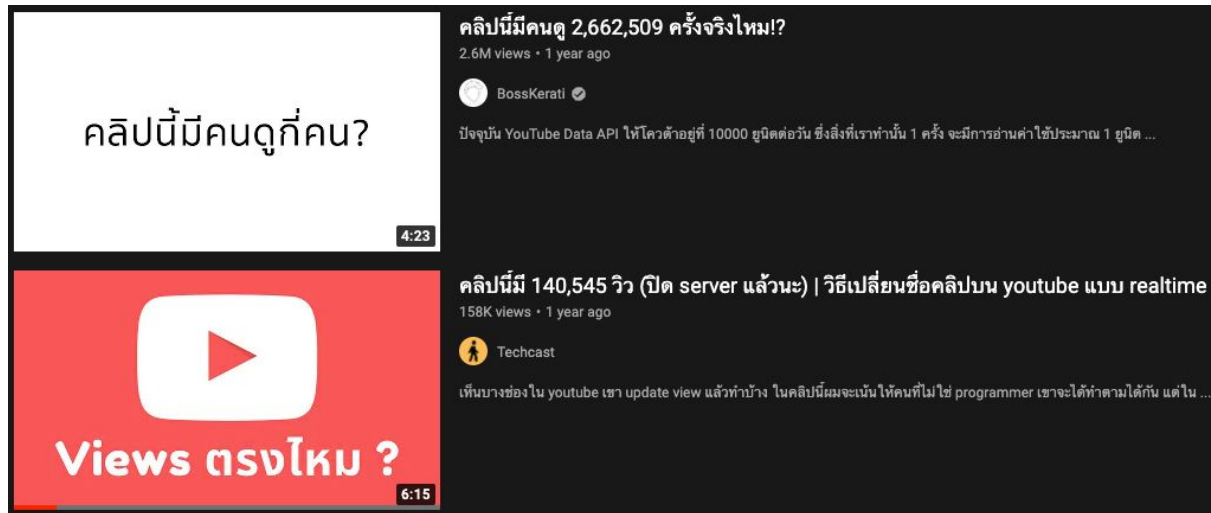
---

Play with it	15 min
--------------	--------

# This video has N views



# This video has N views



# What is “API”?

## Getting Start

**A** - Application

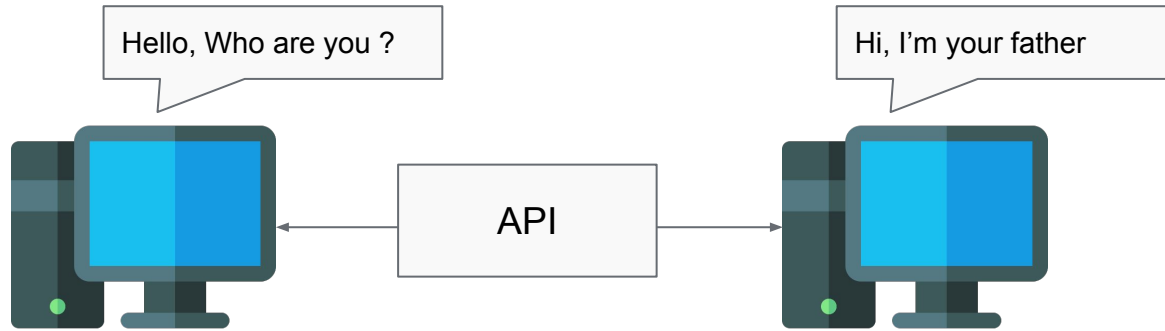
**P** - Programming

**I** - Interface



# How “API” works

Beep Boop



# What is Json?

answer me

- JavaScript Object Notation
- JSON is a text format for storing and transporting data
- JSON is "self-describing" and easy to understand



# JSON Syntax

## How to r/w me

```
{ "key" : "value" }
```

```
{ "name" : "john" }
```

- a string
- a number
- an object
- an array
- a boolean
- null

```
{  
  "name" : "john",  
  "age" : 19,  
  "kids" : [ "Dave", "AJ" ]  
}
```





```
{
  origin: 'Chicago, IL',
  destination: 'Los Angeles, CA',
  waypoints: [
    {
      location: 'Joplin, MO',
      stopover: false
    }, {
      location: 'Oklahoma City, OK',
      stopover: true
    }
  ],
  provideRouteAlternatives: false,
  travelMode: 'DRIVING',
  drivingOptions: {
    departureTime: "time",
    trafficModel: 'pessimistic'
  },
  unitSystem: google.maps.UnitSystem.IMPERIAL
}
```

J  
S  
O  
N

# HTTP Request Anatomy

## components

Verb

Request:

GET /profile HTTP/3

Path

Accept: application/json

Version

Header

Response:

HTTP/3 200 OK

Status

Content-Type: application/json

Body

{ "name" : "john" }



# HTTP Method

action

- **GET** - requests a representation of the specified resource
- **POST** - submits an entity to the specified resource
- **PUT** - replaces all current representations of the target resource with the payload.
- **DELETE** - deletes the specified resource.



# HTTP Status Code

action

- **100** - Informational responses
- **200** - Successful responses
- **300** - Redirection messages
- **400** - Client error responses
- **500** - Server error responses



# HTTP Status Code

Should know

- **400** - Bad Request/Invalid Arguments
- **401** - Unauthenticated (invalid credentials, retry login).
- **403** - Permission denied/forbidden
- **404** - Not found



# TOOLS

try



[Postman](https://www.postman.com/)



[Python Requests](https://requests.readthedocs.io/)



# TOOLS

try



[Cloud Client Libraries | Cloud APIs](#)



# Requests

Python Library





# Installation

Do it

```
python -m pip install requests
```

```
pip install requests
```

```
conda install requests
```

# Coding

Time to do some magic

<https://bit.ly/gdsc-requests>



# Google APIs Walkthrough

walkthrough

## Let's go!!!!

[Google APIs Explorer](#)

### Code Example

[https://colab.research.google.com/drive/1vSeHbtIvXDQV35TrI\\_Lk2MRt-l8zS52q?usp=sharing](https://colab.research.google.com/drive/1vSeHbtIvXDQV35TrI_Lk2MRt-l8zS52q?usp=sharing)



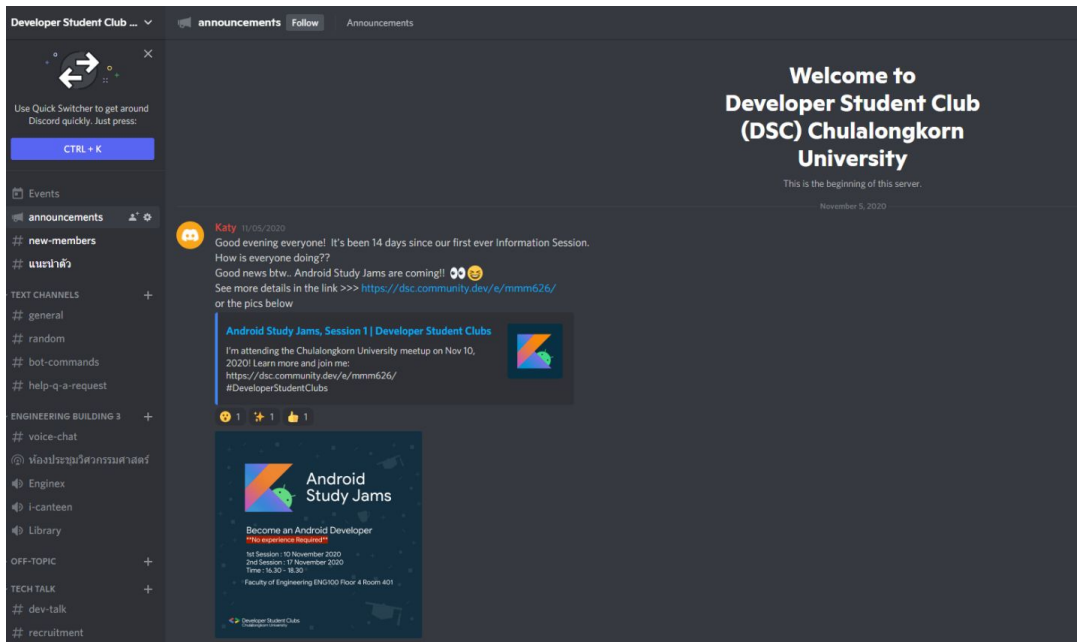
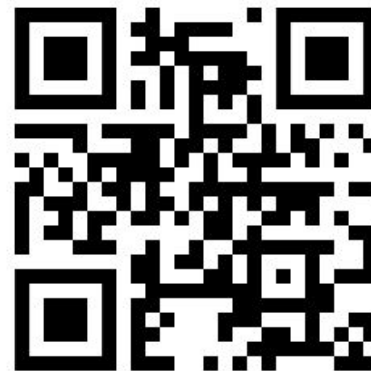
```
1 import os
2
3 import google_auth_oauthlib.flow
4 import googleapiclient.discovery
5 import googleapiclient.errors
6
7 scopes = ["https://www.googleapis.com/auth/youtube.force-ssl"]
8
9
10 def main():
11     os.environ["OAUTHLIB_INSECURE_TRANSPORT"] = "1"
12
13     api_service_name = "youtube"
14     api_version = "v3"
15     client_secrets_file = "./client_secret.json"
16
17     # Get credentials and create an API client
18     flow = google_auth_oauthlib.flow.InstalledAppFlow.from_client_secrets_file(
19         client_secrets_file, scopes)
20     credentials = flow.run_console()
21     youtube = googleapiclient.discovery.build(
22         api_service_name, api_version, credentials=credentials)
23
24     request = youtube.videos().update(
25         part="snippet",
26         body={
27             "id": "1kIqTjaK_0g",
28             "snippet": {
29                 "categoryId": "27",
30                 "title": "Hello Pday"
31             }
32         }
33     )
34     response = request.execute()
35
36     print(response)
37
38
39 if __name__ == "__main__":
40     main()
41
```



```
1  import os
2
3  import google_auth_oauthlib.flow
4  import googleapiclient.discovery
5  import googleapiclient.errors
6
7  scopes = ["https://www.googleapis.com/auth/youtube.readonly"]
8
9
10 def main():
11     os.environ["OAUTHLIB_INSECURE_TRANSPORT"] = "1"
12
13     api_service_name = "youtube"
14     api_version = "v3"
15     client_secrets_file = "client_secret.json"
16
17     flow = google_auth_oauthlib.flow.InstalledAppFlow.from_client_secrets_file(
18         client_secrets_file, scopes)
19     credentials = flow.run_console()
20     youtube = googleapiclient.discovery.build(
21         api_service_name, api_version, credentials=credentials)
22
23     request = youtube.videos().list(
24         part="statistics,snippet",
25         id="1kIqTjaK_0g"
26     )
27     response = request.execute()
28
29     print(response)
30
31
32 if __name__ == "__main__":
33     main()
34
```

# Discord Server

<https://discord.gg/yyCU2XZ>



# Chulalongkorn University



Join Us

## Join us!!

<https://dsc.community.dev/chulalongkorn-university/>



# Q&A Session

Let us help you







**Thank you,**  
**and see you next time!!**

```
function filterStudies({ studies, filterByOrg = false, filterByYear = false }) {  
  return studies.filter(study => {  
    if (filterByOrg) {  
      return study.organization !== 'NIH';  
    }  
    if (filterByYear) {  
      return study.year !== 2019;  
    }  
    return true;  
  });  
}
```