PowerShell Conference Europe

about_VideoProcessing

(Season 2)

Thorsten Butz

PRAGUE23

Thursday, 22 June 2023, 11:00 h
Track 4, PSConfEU

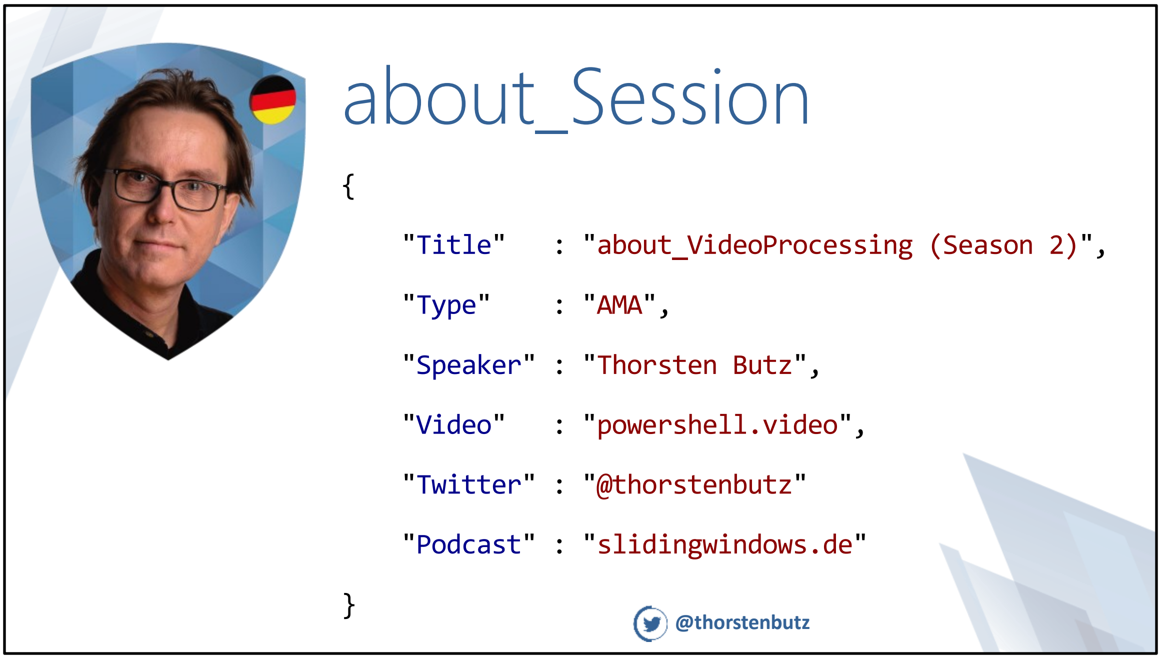
Cubex Centrum Praha
http://www.cubexcentrum.cz/
Na Strži 2097/63, 140 00 Praha 4-Nusle

Many thanks to our sponsors:

PSCONF.EU
PRAGUE23
Chocolatey
Centino systems
PATCH MY PC
ScriptRunner®
The #1 for PowerShell Management

Thanks for **sponsoring** this event!

- https://chocolatey.org
- https://patchmypc.com
- https://www.centinosystems.com
- https://www.scriptrunner.com
- https://synedgy.com



about_Session

```
{
    "Title"   : "about_VideoProcessing (Season 2)",
    "Type"    : "AMA",
    "Speaker" : "Thorsten Butz",
    "Video"   : "powershell.video",
    "Twitter" : "@thorstenbutz"
    "Podcast" : "slidingwindows.de"
}
```

@thorstenbutz

Thorsten Butz is a distinguished "Microsoft Certified Trainer" (MCT), consultant, book author and podcaster. He started his career in the late 1990s as a networking specialist and Unix enthusiast. He conducts Microsoft's server technologies since 2000. His desire for scripting and automation brought him at an early stage to his current focal point: the Windows PowerShell
Thorsten is the host of the "Sliding Windows" (sllidingwindows.de) podcast.

Previously

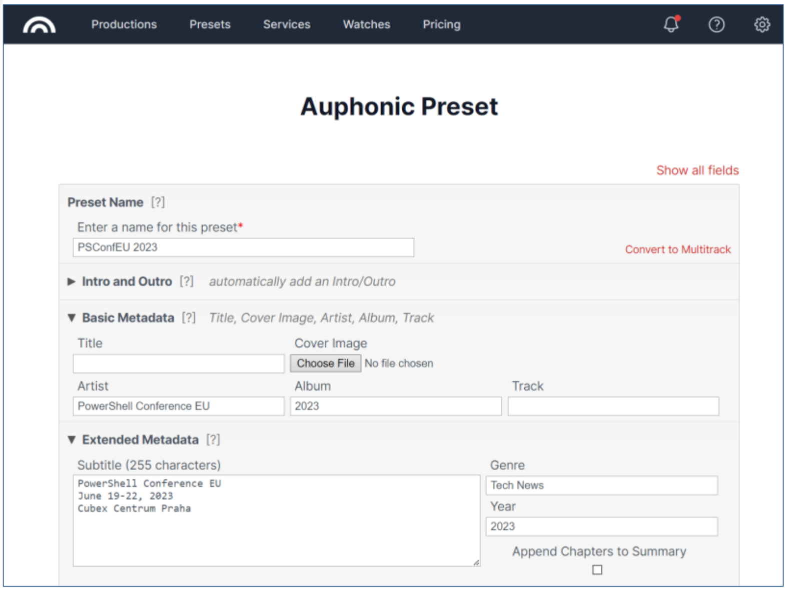
about_VideoProcessing - Thorsten Butz - PSConfEU 2022

Video:

https://youtu.be/i4watr8txOM

Slides and code:

https://github.com/psconfeu/2022/tree/main/ThorstenButz/about_VideoProcessing



Auphonic Preset

Preset Name [?]

Enter a name for this preset*: PSConfEU 2023

▶ **Intro and Outro** [?] *automatically add an Intro/Outro*

▼ **Basic Metadata** [?] *Title, Cover Image, Artist, Album, Track*

Title:

Cover Image: No file chosen

Artist: PowerShell Conference EU

Album: 2023

Track:

▼ **Extended Metadata** [?]

Subtitle (255 characters):

```
PowerShell Conference EU
June 19-22, 2023
Cubex Centrum Praha
```

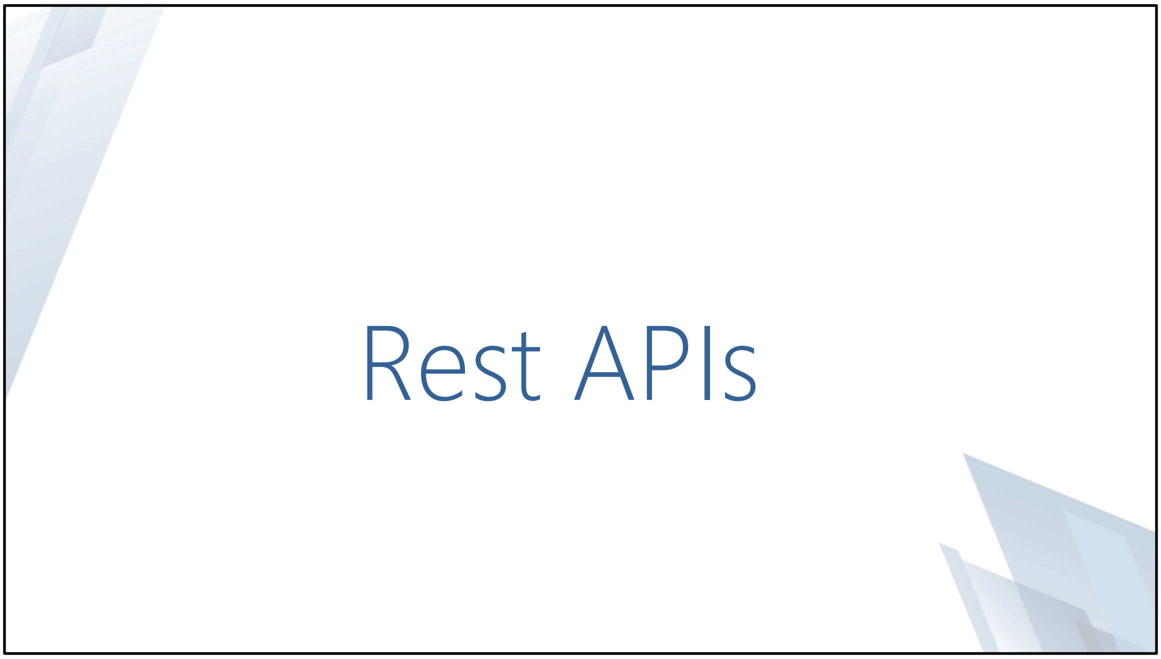
Genre: Tech News

Year: 2023

Append Chapters to Summary ☐

Highly recommended

https://auphonic.com

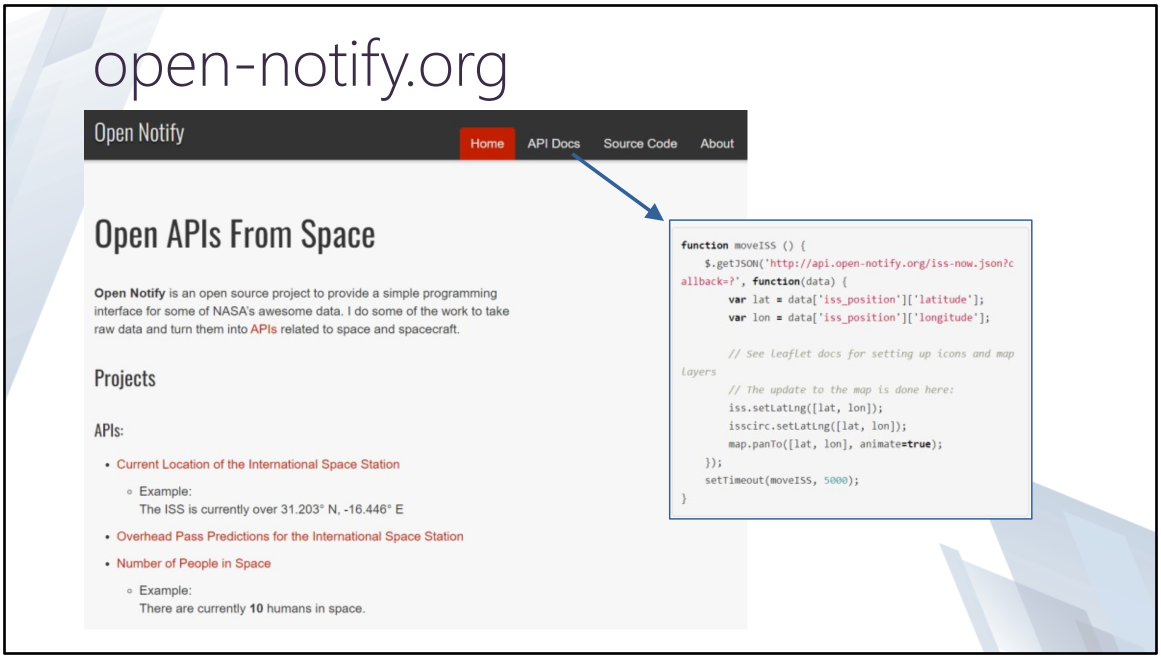


Rest APIs

What is a REST API?

"A REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. REST stands for representational state transfer and was created by computer scientist Roy Fielding."

Reference: https://www.redhat.com/en/topics/api/what-is-a-rest-api



open-notify.org

Open APIs From Space

Open Notify is an open source project to provide a simple programming interface for some of NASA's awesome data. I do some of the work to take raw data and turn them into APIs related to space and spacecraft.

Projects

APIs:

- Current Location of the International Space Station
  - Example:
    The ISS is currently over 31.203° N, -16.446° E
- Overhead Pass Predictions for the International Space Station
- Number of People in Space
  - Example:
    There are currently **10** humans in space.

```
function moveISS () {
    $.getJSON('http://api.open-notify.org/iss-now.json?c
allback=?', function(data) {
        var lat = data['iss_position']['latitude'];
        var lon = data['iss_position']['longitude'];

        // See leaflet docs for setting up icons and map
layers
        // The update to the map is done here:
        iss.setLatLng([lat, lon]);
        isscirc.setLatLng([lat, lon]);
        map.panTo([lat, lon], animate=true);
    });
    setTimeout(moveISS, 5000);
}
```

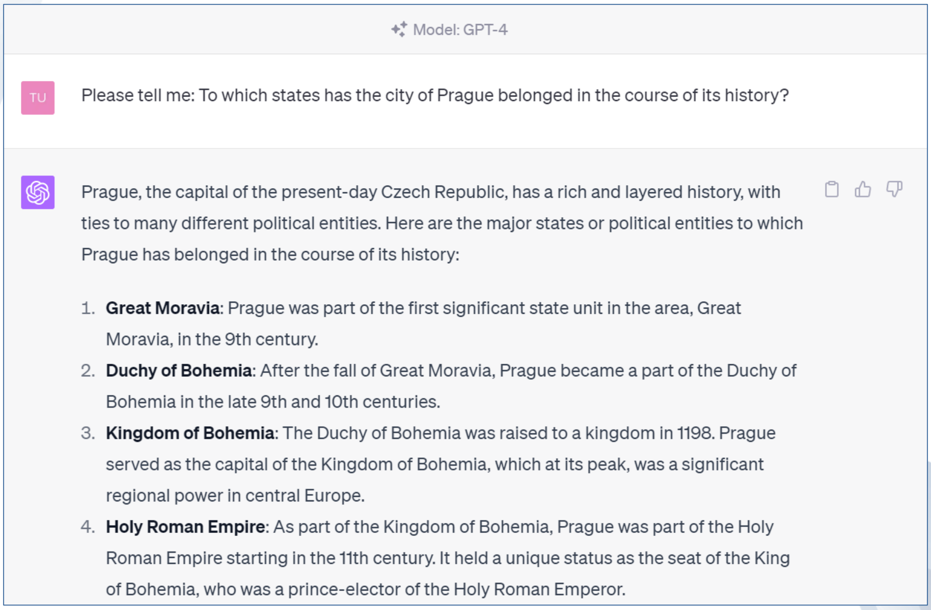
```
## A
$request = Invoke-RestMethod -Uri 'http://api.open-
notify.org/astros.json'
$request.people

## B
function Get-PeopleSpace {
    param ($craft)
    $request = Invoke-RestMethod -Uri 'http://api.open-
notify.org/astros.json'
    if ($craft) {
        $request.people | Where-Object -FilterScript { $_.craft
-eq $craft}
    }
    else {
        $request.people
    }
}

Get-PeopleSpace -craft ISS
```

AMA

Ask me anything ..

A KI A

Ask the KI anything ..



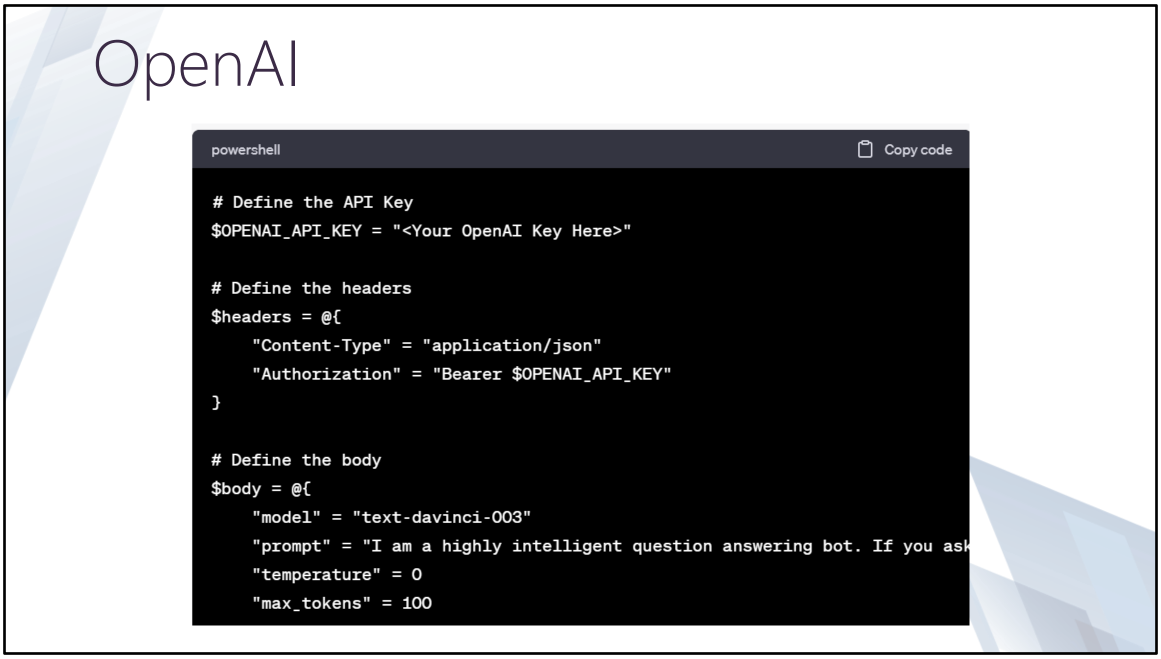
Please tell me: To which states has the city of Prague belonged in the course of its history?

Prague, the capital of the present-day Czech Republic, has a rich and layered history, with ties to many different political entities. Here are the major states or political entities to which Prague has belonged in the course of its history:

1. **Great Moravia**: Prague was part of the first significant state unit in the area, Great Moravia, in the 9th century.
2. **Duchy of Bohemia**: After the fall of Great Moravia, Prague became a part of the Duchy of Bohemia in the late 9th and 10th centuries.
3. **Kingdom of Bohemia**: The Duchy of Bohemia was raised to a kingdom in 1198. Prague served as the capital of the Kingdom of Bohemia, which at its peak, was a significant regional power in central Europe.
4. **Holy Roman Empire**: As part of the Kingdom of Bohemia, Prague was part of the Holy Roman Empire starting in the 11th century. It held a unique status as the seat of the King of Bohemia, who was a prince-elector of the Holy Roman Emperor.

1. **Great Moravia (833–907)**: Prague was part of the first significant state unit in the area.
2. **Duchy of Bohemia (c. 870–1198)**: Prague became a part of the Duchy of Bohemia after the fall of Great Moravia.
3. **Kingdom of Bohemia (1198–1526)**: The Duchy of Bohemia was raised to a kingdom.
4. **Habsburg Monarchy / Austrian Empire (1526–1867)**: The Habsburgs came into control of the Bohemian crown, incorporating it into their vast territories.
5. **Austria-Hungary (1867–1918)**: The Austrian Empire reorganized into the dual monarchy of Austria-Hungary.
6. **Czechoslovakia (1918–1939)**: After the collapse of Austria-Hungary, Prague became the capital of the newly formed state of Czechoslovakia.
7. **Nazi Germany (1939–1945)**: During World War II, Prague was occupied by Nazi Germany as the capital of the Protectorate of Bohemia and Moravia.
8. **Czechoslovakia (1945–1993)**: After World War II, Prague returned to being the capital of Czechoslovakia, which became a Communist state in 1948 until the Velvet Revolution in 1989.
9. **Czech Republic (1993–Present)**: After the peaceful Velvet Divorce in 1993, Prague has been the capital of the independent Czech Republic.

https://chat.openai.com/share/bf236e91-0e3e-459a-91ea-6f75fb16b4bb

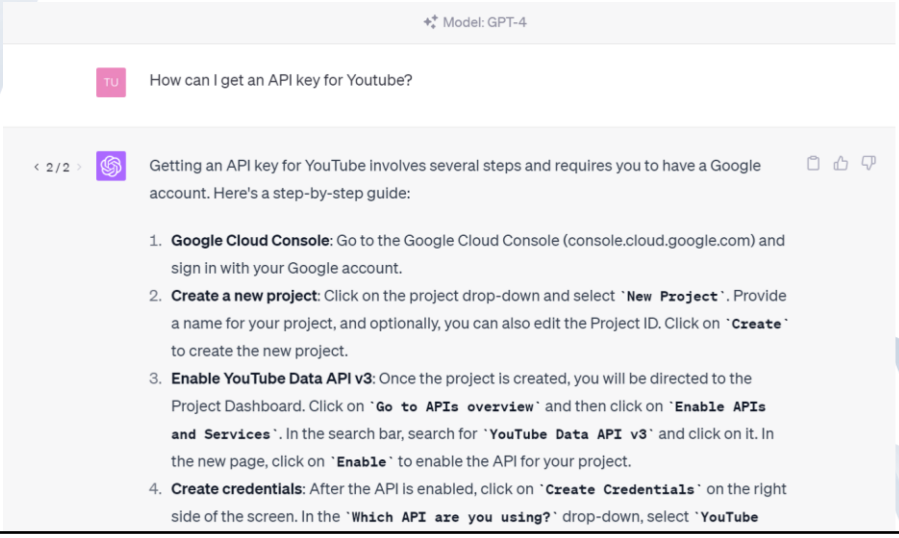


OpenAI

```powershell
# Define the API Key
$OPENAI_API_KEY = "<Your OpenAI Key Here>"

# Define the headers
$headers = @{
    "Content-Type" = "application/json"
    "Authorization" = "Bearer $OPENAI_API_KEY"
}

# Define the body
$body = @{
    "model" = "text-davinci-003"
    "prompt" = "I am a highly intelligent question answering bot. If you ask
    "temperature" = 0
    "max_tokens" = 100
```

```powershell
$OPENAI_API_ORG =
$OPENAI_API_KEY =

## OpenAI Example A
$headers = @{
  'Content-Type' = 'application/json'
  'Authorization' = "Bearer $OPENAI_API_KEY"
  'OpenAI-Organization' =  $OPENAI_API_ORG
}
$request = Invoke-RestMethod -Headers $headers -Uri
'https://api.openai.com/v1/models'
$request.data.id
```

YouTube ꝏ OpenAI



How can I get an API key for Youtube?

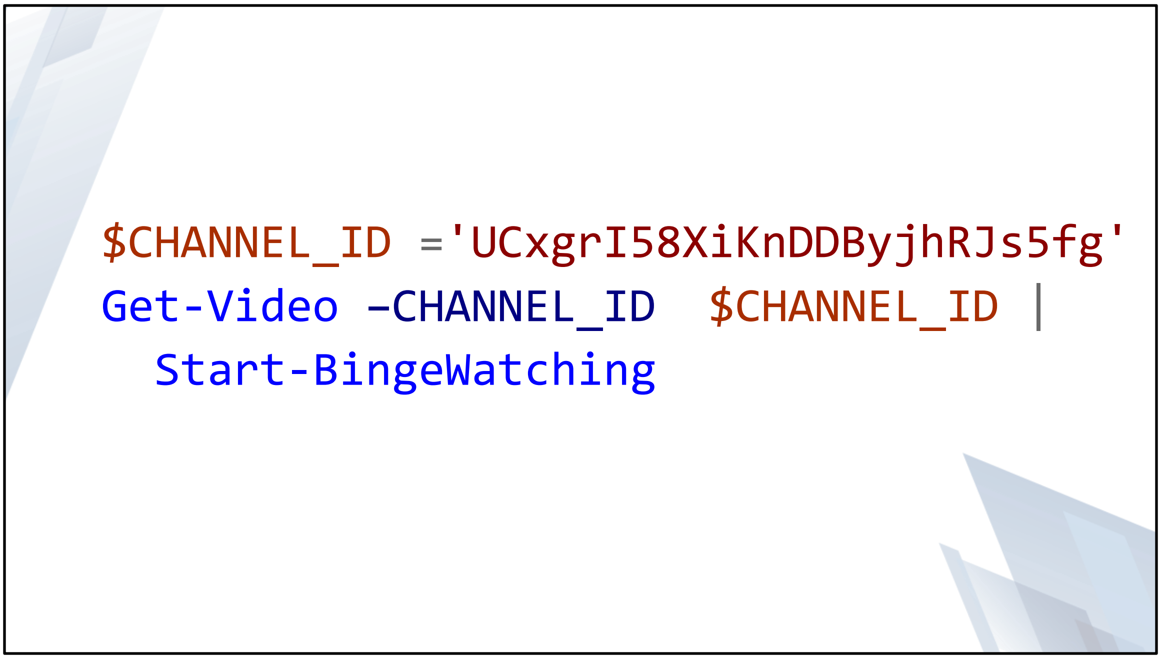
Getting an API key for YouTube involves several steps and requires you to have a Google account. Here's a step-by-step guide:

1. **Google Cloud Console**: Go to the Google Cloud Console (console.cloud.google.com) and sign in with your Google account.
2. **Create a new project**: Click on the project drop-down and select `New Project`. Provide a name for your project, and optionally, you can also edit the Project ID. Click on `Create` to create the new project.
3. **Enable YouTube Data API v3**: Once the project is created, you will be directed to the Project Dashboard. Click on `Go to APIs overview` and then click on `Enable APIs and Services`. In the search bar, search for `YouTube Data API v3` and click on it. In the new page, click on `Enable` to enable the API for your project.
4. **Create credentials**: After the API is enabled, click on `Create Credentials` on the right side of the screen. In the `Which API are you using?` drop-down, select `YouTube

How can I get an API key for Youtube?

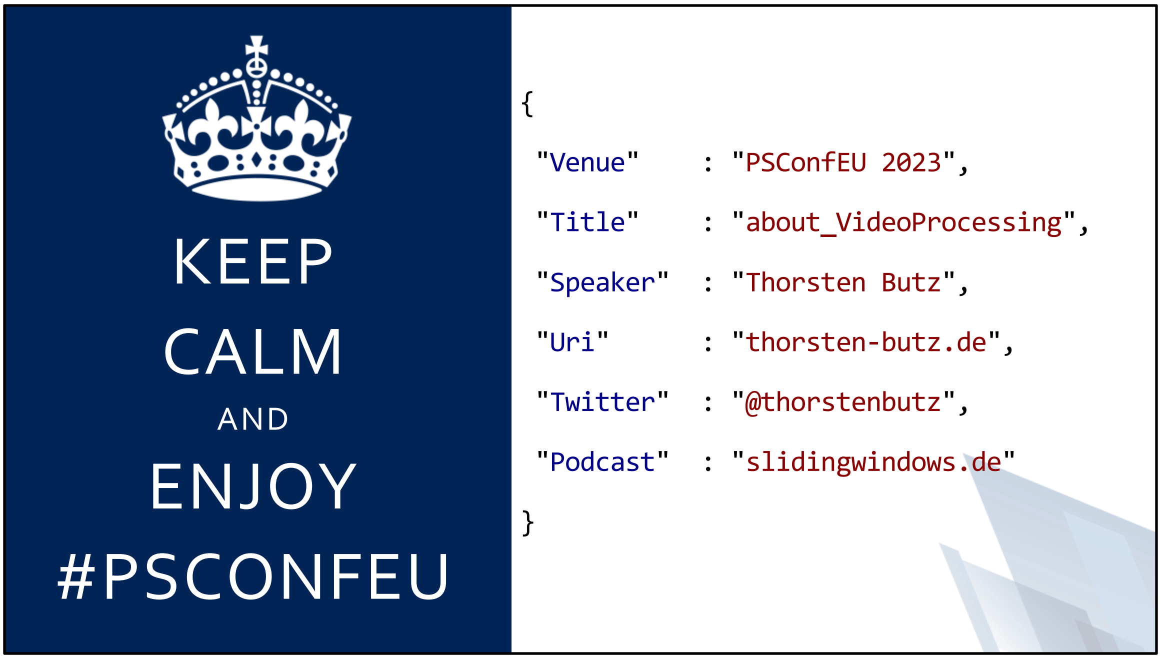
Getting an API key for YouTube involves several steps and requires you to have a Google account. Here's a step-by-step guide:
1.**Google Cloud Console**: Go to the Google Cloud Console (console.cloud.google.com) and sign in with your Google account.
2.**Create a new project**: Click on the project drop-down and select New Project. Provide a name for your project, and optionally, you can also edit the Project ID. Click on Create to create the new project.
3.**Enable YouTube Data API v3**: Once the project is created, you will be directed to the Project Dashboard. Click on Go to APIs overview and then click on Enable APIs and Services. In the search bar, search for YouTube Data API v3 and click on it. In the new page, click on Enable to enable the API for your project.
4.**Create credentials**: After the API is enabled, click on Create Credentials on the right side of the screen. In the Which API are you using? drop-down, select YouTube Data API v3. For Where will you be calling the API from?, select the relevant option (usually Web server (e.g., node.js, Tomcat) if you're going to use it on a server, or Web browser (JavaScript) if it's client-side). Then select Public data under What data will you be accessing?. Then click Next.
5.**API key**: Now, you should be presented with an API key. Copy this key and store it in a safe place.

https://chat.openai.com/share/3b211a18-4d2a-4e1d-ba32-3173284b8bf5



```
$CHANNEL_ID ='UCxgrI58XiKnDDByjhRJs5fg'
Get-Video -CHANNEL_ID  $CHANNEL_ID |
  Start-BingeWatching
```

```
##################################
## Total NO of videos published
##################################

$UriSearch = "https://www.googleapis.com/youtube/v3/search"
$UriChannel = "https://www.googleapis.com/youtube/v3/channels"
$ApiKey = $API_KEY
$ChannelId = $CHANNEL_ID

$ParamsChannel = @{
    key = $ApiKey
    id = $ChannelId
    part = "statistics"
}

$responseChannel = Invoke-RestMethod -Uri $UriChannel -Method
Get -Body $ParamsChannel
$videoCount = $responseChannel.items.statistics.videoCount

Write-Output "Total number of published videos: $videoCount"
```



KEEP

CALM

AND

ENJOY

#PSCONFEU

```
{
 "Venue"     : "PSConfEU 2023",
 "Title"     : "about_VideoProcessing",
 "Speaker"   : "Thorsten Butz",
 "Uri"       : "thorsten-butz.de",
 "Twitter"   : "@thorstenbutz",
 "Podcast"   : "slidingwindows.de"
}
```

```
Start-Broadcasting -yourself
```