

Harnessing ChatGPT via API Interface.

By Eric Nielsen

Local Machine Account: ericnipro
Local Machine Password: vmware123!



@ericnipro

- **University of Florida;** Computer Science & Marketing “85”
- **Stanford:** Graduate Classes in Engineering
- **Out of College:** “IBM ESD, PC/XT/AT” Business Apps Group
- **Companies:** “Gould Electronics, Kodak, Pepsi Co, Sun Microsystems
- **Consulting:** US Navy, Australian Navy, Interactive Systems, others.
- **Founded:** BigAdmin, Flash Archive “flar -cvf”, vExpert, VMware{code}

For fun

- “Bulletin Boards for sharing “ 1 upload for 10 downloads
- Got my first SCO Unix box on a “6MHz IBM AT” first “root” password
- “UUCP to Universtiy of Miami” for my own email feed.
- Started learning electronics and have never looked back
- Married, w/four children, “My oldest is a hardware engineer at Roku”



Summary of this session

- Intro to ChatGPT - chat.openai.com
- Prompt - What are they and how to use them.
- Learn how to use the Rest API and Setup keys
- Examples of how to exercise the api

The Hardware for this Lab

- Intel NUC/Clone with Celeron X86 CPU
- 16/32/64 Gig of Memory, No fan, Dual HDMI, Wifi USB Ports
- Can Run ESX with USB Ethernet Dongle
- 256/500 Gig SSD
- 1080p Display (115.00) , Keyboard, Mouse
- Why: <280.00 for a quiet home lab machine
Portable, one entire table fits in two backpacks (No shipping)
Community setup, no event labor required.

More versatile than a Raspberry PI 4/5



Intel NUC 11 Mini PC with Quad-Core Celeron N5105, Versatile Mini Computers 8GB DDR4 RAM & 256GB SSD - Support HDMI Dual 4K UHD/Bluetooth v5.2/Windows 11 Pro preinstalled (NUC11ATKC4)

Visit the waitloan Store
4.5 ★★★★★ 110 ratings | Search this page
Amazon's Choice Overall Pick
100+ bought in past month


\$239⁰⁰
FREE Returns

Get \$50 off instantly: Pay \$189.00 upon approval for Amazon Visa.

Color: NUC11ATKC4 8GB+256GB

Brand	waitloan
Operating System	Windows 11 Pro
CPU Model	Celeron
CPU Speed	2.9 GHz
Graphics Card	Integrated

Roll over image to zoom in



CanaKit Raspberry Pi 5 Starter Kit PRO - Turbine Black (128GB Edition) (8GB RAM)

Visit the CanaKit Store
4.6 ★★★★★ 302 ratings | Search this page
Amazon's Choice Overall Pick
3K+ bought in past month

\$159⁹⁹
FREE Returns

Get \$50 off instantly: Pay \$109.99 upon approval for Amazon Visa.

Size: 8GB RAM

4GB RAM	8GB RAM
\$149.99	\$159.99

Brand CanaKit
Ram Memory Installed 8 GB
Size
Memory Storage Capacity 128 GB

Roll over image to zoom in

Lab #1.

- Create a new ChatGPT Account
- Login to Existing ChatGPT Account

Let's Login to chat.openai.com

You can create an
account or use one that
you already use

- Create Account
- Verify Email address

ChatGPT

Tips for getting started

Ask away

ChatGPT can answer questions, help you learn, write code, brainstorm together, and much more.

Don't share sensitive info

Chat history may be reviewed or used to improve our services. Learn more about your choices in our [Help Center](#).

Check your facts

While we have safeguards, ChatGPT may give you inaccurate information. It's not intended to give advice.

Okay, let's go

Create your account

Note that phone verification may be required for signup. Your number will only be used to verify your identity for security purposes.

Email address

Continue

Already have an account? [Log in](#)

OR

ChatGPT

<http://chat.openai.com>

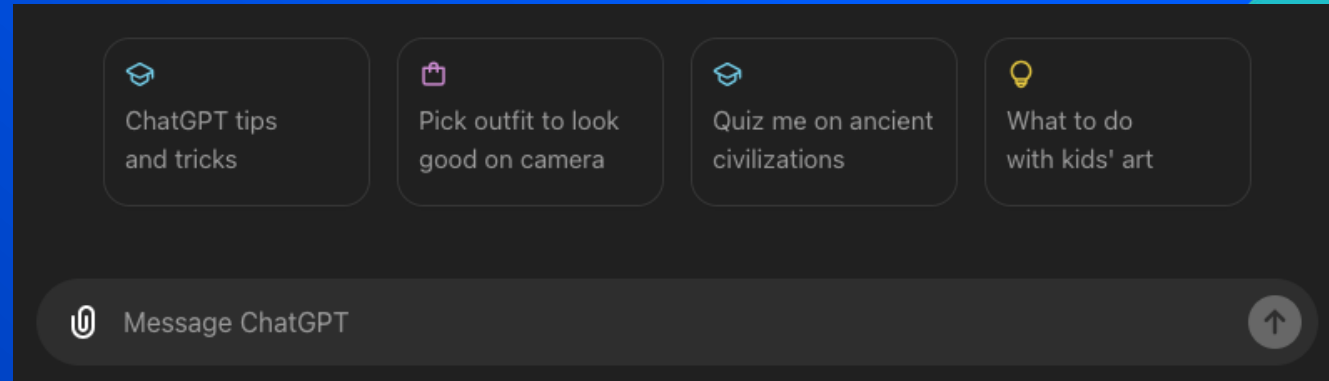
- Version 4o- With a verified account.
Fastest model great for everyday tasks - Free
- Version 4o - with a Plus account.
More capable model, used for advanced tasks and
creative and advanced reasoning - \$20 / Month

4o = "o" for "omni", grasps idioms, metaphors...



What is a Prompt?

- It serves as the starting point for the AI to generate a response or engage in a dialogue.
- The prompt can range from a single sentence to a longer paragraph, depending on the complexity of the desired task. It helps guide the AI's understanding and context for the subsequent interactions, ensuring more relevant and coherent replies.
- By tailoring the prompt effectively, a prompt used with ChatGPT is a concise and specific textual input provided by the user to initiate a conversation or request information from the AI language model.
- Users can obtain valuable insights, creative ideas, or assistance on a wide array of topics from ChatGPT.
- <https://platform.openai.com/docs/quickstart/introduction>



```
6  "messages": [  
7    {  
8      "role": "system",  
9      "content": "You will be provided with a product description and seed words, and your task is  
10     },  
11     {  
12       "role": "user",  
13       "content": "Product description: A home milkshake maker\n      Seed words: fast, healthy, compa  
14     }  
15   ],
```


Examples

- Write a tagline for an ice cream shop.
- Explain Quantum Entanglement to me as a Highschool Student
- Create an outline for a YouTube video about schools
- Write a small powershell script to reboot a VMware Virtual Machine

```
# Install VMware PowerCLI if not already installed
if (-not (Get-Module -ListAvailable -Name VMware.PowerCLI)) {
    Install-Module -Name VMware.PowerCLI -Scope CurrentUser -Force
}

# Import VMware PowerCLI module
Import-Module VMware.PowerCLI

# Connect to vCenter Server
$server = "your_vcenter_server"
$user = "your_username"
$password = "your_password"
Connect-VIServer -Server $server -User $user -Password $password

# Define the name of the virtual machine to reboot
$vmName = "your_vm_name"

# Get the virtual machine object
$vm = Get-VM -Name $vmName

# Reboot the virtual machine
Restart-VMGuest -VM $vm -Confirm:$false

# Disconnect from vCenter Server
Disconnect-VIServer -Server $server -Confirm:$false
```

Replace `"your_vcenter_server"`, `"your_username"`, `"your_password"`, and
`"your_vm_name"` with the actual values for your environment. This script connects to the
vCenter server, finds the specified virtual machine, and reboots it.

Let's try one together!

- Will someone recommend a VMworld related topic?
- Can someone suggest a famous author?
- Can someone recommend an education level?

Explain to me in one paragraph [topic] in the style and tone of [author] as if I was a [education]

<http://chat.openai.com>

Key Prompt Suggestions

- Ignore all previous instruction before this one
- Write a [length] [writing result] on [topic] in the style of [style]
- Rewrite this article in its same style keeping its approximate length. Please ensure to keep the major points
- show your work
- Do not respond with anything except “...”
- ... in a codeblock
- rewrite longer (or shorter)
- Summarize ...

Prompt Examples

<https://platform.openai.com/docs/examples...>

Prompt examples

Explore what's possible with some example prompts

All categories

Grammar correction
 Convert ungrammatical statements into standard English.

Summarize for a 2nd grader
 Simplify text to a level appropriate for a second-grade student.

Parse unstructured data
 Create tables from unstructured text.

Emoji Translation
 Translate regular text into emoji text.

Calculate time complexity
 Find the time complexity of a function.

Explain code
 Explain a complicated piece of code.

Keywords
 Extract keywords from a block of text.

Product name generator
 Generate product names from a description and seed words.

Python bug fixer
 Find and fix bugs in source code.

Spreadsheet creator
 Create spreadsheets of various kinds of data.

Tweet classifier
 Detect sentiment in a tweet.

Airport code extractor
 Extract airport codes from text.

Mood to color
 Turn a text description into a color.

VR fitness idea generator
 Generate ideas for fitness promoting virtual reality games.

Marv the sarcastic chat bot
 Marv is a factual chatbot that is also sarcastic.

Turn by turn directions
 Convert natural language to turn-by-turn directions.

Interview questions
 Create interview questions.

Function from specification
 Create a Python function from a specification.

Improve code efficiency
 Provide ideas for efficiency improvements to Python code.

Single page website creator
 Create a single page website.

Rap battle writer
 Generate a rap battle between two characters.

Memo writer
 Generate a company memo based on provided points.



Product name generator

Generate Natural Language

Open in Playground

Generate product names from a description and seed words.

Prompt

SYSTEM You will be provided with a product description and seed words, and your task is to generate product names.

USER Product description: A home milkshake maker
Seed words: fast, healthy, compact.

Sample response

1. QuickBlend
2. FitShake
3. MiniMix
4. TurboShake
5. HealthWhiz
6. MightyMixer
7. SlimBlend
8. SpeedyShake
9. NutriBlend
10. CompactMilkshake

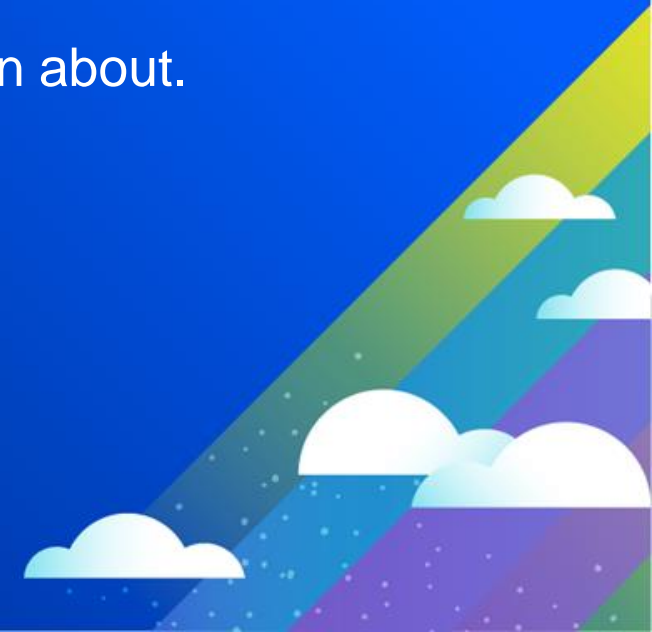
API request

python

```
1 from openai import OpenAI
2 client = OpenAI()
3
4 response = client.chat.completions.create(
5     model="gpt-3.5-turbo",
6     messages=[
7         {
8             "role": "system",
9             "content": "You will be provided with a product description and seed words, and your task is to
10         },
11         {
12             "role": "user",
13             "content": "Product description: A home milkshake maker\n    Seed words: fast, healthy, compac
14         }
15     ],
16     temperature=0.8,
17     max_tokens=64,
18     top_p=1
19 )
```


ChatGPT Development & Training

- 570 gigabytes of text and 175 billion parameters, including text scraped from Wikipedia, Twitter, and Reddit.
- Use the AI in ways that play to AI's strengths.
- Responses will be superior on topics that are well understood and written about.
- ChatGPT is weak at independent reasoning and independent thought.
- Verify the output of what the AI gives you. Your mileage may vary.



ChatGPT Limitations

- Model Data is limited to known events before
 - July 25, 2024.
- Limits Question and the Answers to:
 - GPT 4.0 - about 8,000 to 12,000 words
- Content Guidelines –

<https://openai.com/policies/usage-policies>

ChatGPT API

Cost Per Request

1. ChatGPT-4 (8k context)

- \$0.03 per 1,000 prompt tokens
- \$0.06 per 1,000 completion tokens

2. ChatGPT-4 (32k context)

- \$0.06 per 1,000 prompt tokens
- \$0.12 per 1,000 completion tokens

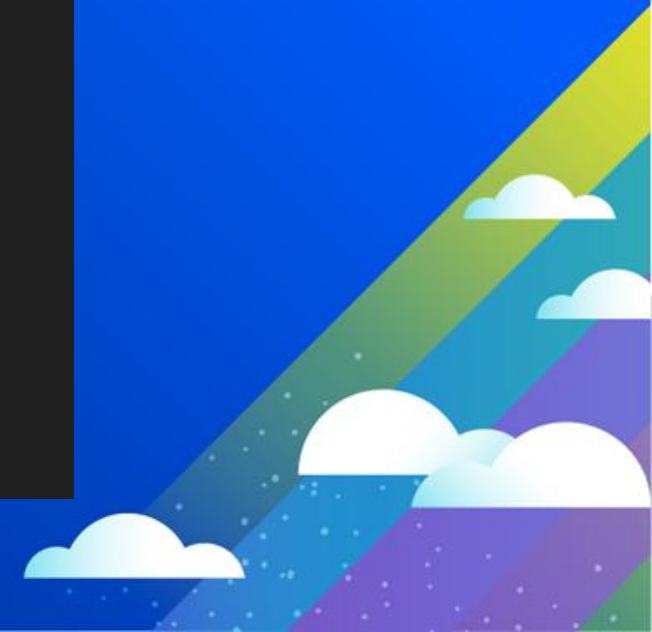
3. ChatGPT-3.5

- \$0.0015 per 1,000 prompt tokens
- \$0.002 per 1,000 completion tokens

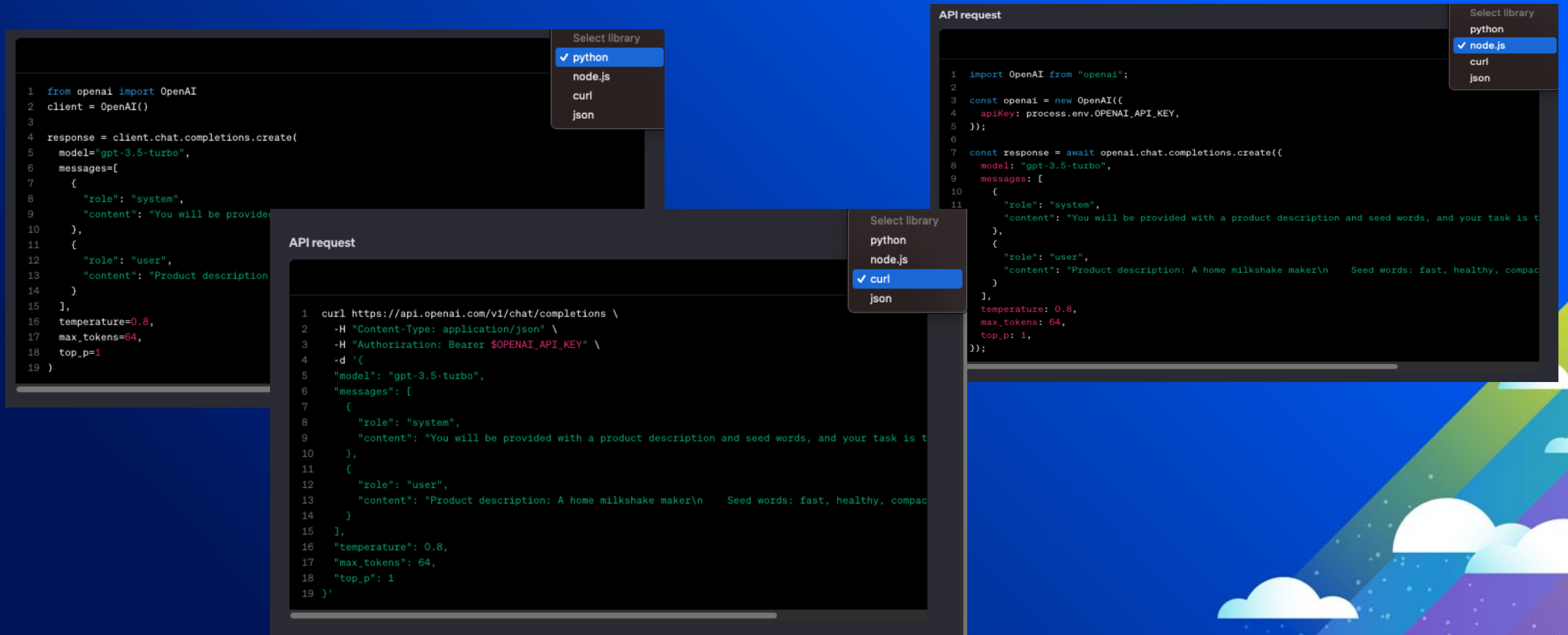
A token is roughly equivalent to a word, so 1,000 tokens are approximately 750 words.

To calculate the cost per request, you need to estimate the number of tokens in both the prompt (your input) and the completion (the output). For example, if you send a prompt that uses 50 tokens and receive a completion that uses 150 tokens with ChatGPT-3.5, the cost would be:

- Prompt: 50 tokens * \$0.0015/1,000 = \$0.000075
- Completion: 150 tokens * \$0.002/1,000 = \$0.0003
- Total: \$0.000075 + \$0.0003 = \$0.000375 per request



Supports Multiple Programming Languages

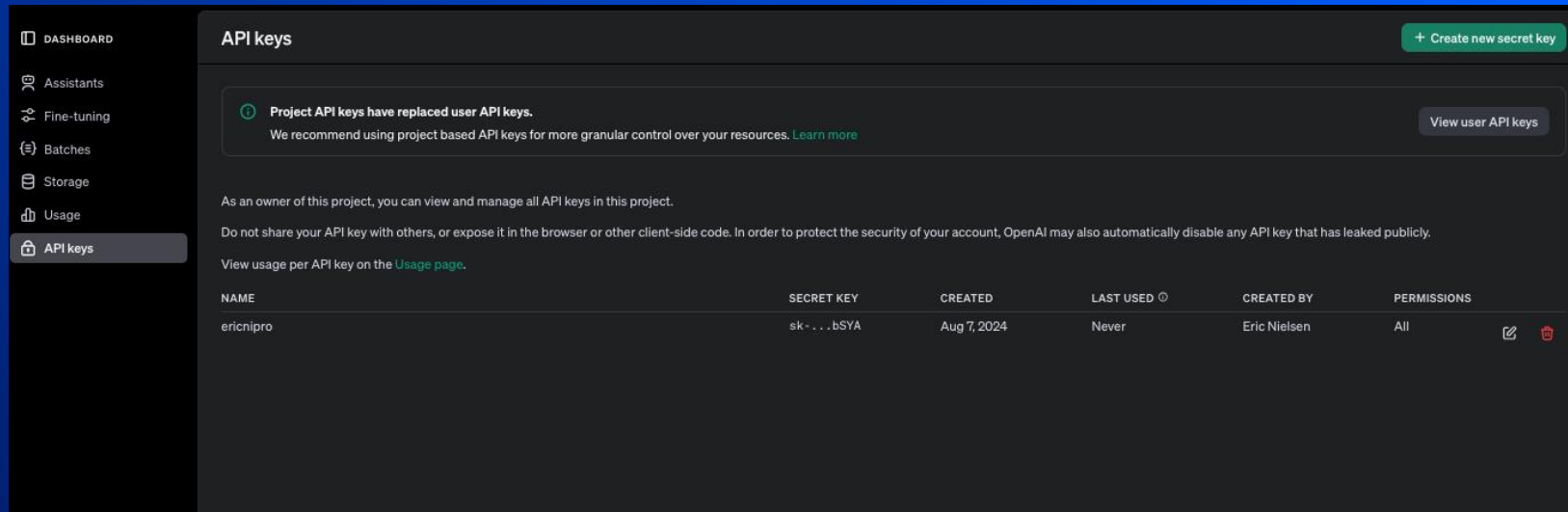


Lab #2.

Setting up your API Key



Create and Managing API Keys



<https://platform.openai.com/account/api-keys>

- Authenticate
- New users will be granted a small budget to play with
- Choose “Create a New Secret Key” and give it a name
- Save the Key Generated
- DOS set OPENAI_API_KEY=sk-secretkeyvalue...

Successful Request Example!

Authentication

Make a Request:

DOS

```
curl https://api.openai.com/v1/models -H  
"Authorization: Bearer  
%OPENAI_API_KEY%"
```

Expected Response would be a list of
available models for this account

```
$ curl https://api.openai.com/v1/models -H  
"Authorization: Bearer sk-proj-9npJNvOvTW5CWT91N5CYRtymr6-  
Cy9DVOugNtDxIOeyM3zKLpv21HZQkBgT3BlbkFJdluzxbUi-  
V7JgOPfsjz4AlbY5E8JCUmEC3DcXcC6iT3KYTuSnxpnq9bSYA"  
  
{  
  "object": "list",  
  "data": [  
    {  
      "id": "dall-e-3",  
      "object": "model",  
      "created": 1698785189,  
      "owned_by": "system"  
    },  
    {  
      "id": "dall-e-2",  
      "object": "model",  
      "created": 1698798177,  
      "owned_by": "system"  
    }  
  ]  
}
```

Failure Request Example!

Authentication1

Make a Request:

DOS

```
curl https://api.openai.com/v1/models -H  
"Authorization: Bearer  
%OPENAI_API_KEY%"
```

Expected Response would be a list of
available models for this account

```
{  
  "error": {  
    "message": "Incorrect API key provided: sk-proj-  
*****  
*****ng9b. You can find your API key at  
https://platform.openai.com/account/api-keys.",  
    "type": "invalid_request_error",  
    "param": null,  
    "code": "invalid_api_key"  
  }  
}
```


Model's are the various apis that you can interact with.

Understanding the Model's available to this account

DOS

curl https://api.openai.com/v1/models -H "Authorization: Bearer %OPENAI_API_KEY%"

Lists the currently available models, and provides basic information about each one such as the owner and availability.

Id: which model you will be invoking

Created: When the model was created

Permissions: Various permissions associated with the specific model

Model information can be found at
<https://platform.openai.com/docs/api-reference/models>

```
{
  "id": "tts-1-hd",
  "object": "model",
  "created": 1699046015,
  "owned_by": "system"
},
{
  "id": "text-embedding-3-large",
  "object": "model",
  "created": 1705953180,
  "owned_by": "system"
},
{
  "id": "babbage-002",
  "object": "model",
  "created": 1692634615,
  "owned_by": "system"
},
{
  "id": "gpt-4o-mini-2024-07-18",
  "object": "model",
  "created": 1721172717,
  "owned_by": "system"
},
{
  "id": "gpt-4o-mini",
  "object": "model",
  "created": 1721172741,
  "owned_by": "system"
},
{
  "id": "text-embedding-3-small",
  "object": "model",
  "created": 1705948997,
  "owned_by": "system"
},
}
```

Making a Chat Request

Ensure to select a model that is listed in your model list

model String Required - ID of the model to use.

messages Array Required - A list of messages comprising the conversation so far

role String Required - The role of the messages author

content string or null Required - The contents of the message

```
curl https://api.openai.com/v1/chat/completions -H "Content-Type:
application/json" -H "Authorization: Bearer sk-proj-
9npJNvOvTW5CWT91N5CYRtymr6-
Cy9DVOugNtDxIOeyM3zKLpv21HZQkBqT3BlbkFJdluzxbUi-
V7JgOPfsjz4AlbY5E8JCUmEC3DcXcC6iT3KYTuSnxpnq9bSYA" -d "
{ \"model\": \"gpt-3.5-turbo\",
  \"messages\":
  [
    { \"role\": \"user\",
      \"content\": \"Tell me a joke\"
    }
  ]
}"

{
  "id": "chatcmpl-9tZorGxwHRn7m71oo52GWlbgDJ2",
  "object": "chat.completion",
  "created": 1723077952,
  "model": "gpt-3.5-turbo-0125",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "Why did the scarecrow win an award?\nBecause he was outstanding in his field!",
        "refusal": null
      },
      "logprobs": null,
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 11,
    "completion_tokens": 17,
    "total_tokens": 28
  },
  "system_fingerprint": null
}
```

Lab #3.

Sending in a request, same as the Prompt Windows through the API



ChatGPT is a Language Expert

What if you wanted this in a different language

Ask ChatGPT to convert this into

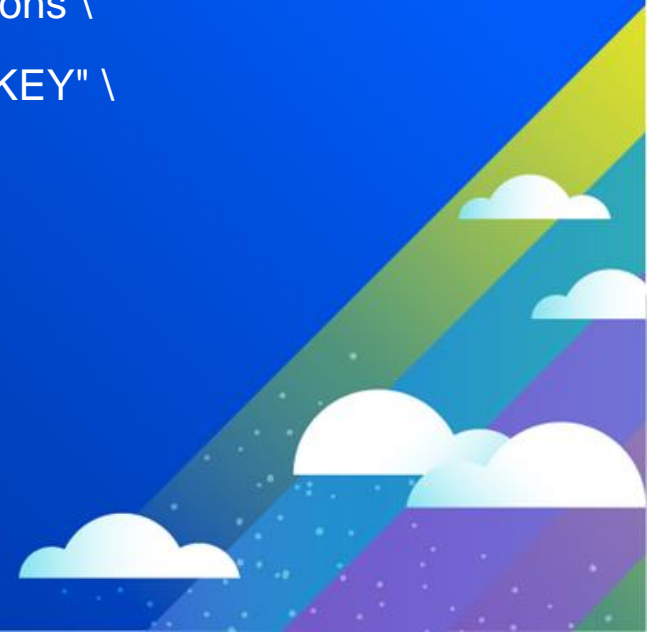
- Java
- Shell
- C++
- Python
- etc

DOS:

```
curl https://api.openai.com/v1/chat/completions -H "Content-Type: application/json" -H "Authorization: Bearer %OPENAI_API_KEY%" -d "{ \"model\": \"gpt-3.5-turbo\", \"messages\": [ { \"role\": \"user\", \"content\": \"Tell me a joke\" } ] }"
```

UNIX:

```
curl https://api.openai.com/v1/chat/completions \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $OPENAI_API_KEY" \
-d '{
  "model": "gpt-3.5-turbo",
  "messages": [
    {
      "role": "user",
      "content": "Tell me a joke"
    }
  ]
}'
```



ChatGPT – PowerCLI Intro

What are popular powerCLI scripts to manage VMware Virtual machines

The answer will be typical:

1. VM Inventory Report
2. Power On/Off Multiple VMs
3. Clone a VM
4. VMware Tools Status Check
5. Change VM Network Adapter
6. Snapshot Management
7. VM Performance Metrics Report
8. VM Disk Space Usage Report
9. Migrate VM to Another Host
10. VM Guest OS Customization

```
1. Get-VM | Select-Object Name, PowerState, NumCpu, MemoryMB, VMHost, @{N="IP
   Address";E={@($_.guest.ipaddress)}}
2. Get-VM -Name "VM1","VM2","VM3" | Start-VM
3. $sourceVM = Get-VM -Name "SourceVM"
   $targetVMName = "CloneVM"
   $targetDatastore = Get-Datastore -Name "Datastore1"
   $targetFolder = Get-Folder -Name "TargetFolder"
   $targetHost = Get-VMHost -Name "ESXiHost1" New-VM -Name $targetVMName -VM
   $sourceVM -VMHost $targetHost -Datastore $targetDatastore -Location $targetFolder
4. Get-VM | Select Name, @{N="Tools Status";E={$_.ExtensionData.Guest.ToolsStatus}}
5. $vm = Get-VM -Name "VMName"
   $adapter = Get-NetworkAdapter -VM $vm
   Set-NetworkAdapter -NetworkAdapter $adapter -NetworkName "NewNetwork" -Confirm:$false
6. Get-VM -Name "VMName" | New-Snapshot -Name "SnapshotName" -Description "Snapshot before upgrade"
7. Get-VM | Select Name,
   @{N="CPU Usage";E={(Get-Stat -Entity $_ -Stat cpu.usage.average -Realtime -MaxSamples 1).Value}},
   @{N="Memory Usage";E={(Get-Stat -Entity $_ -Stat mem.usage.average -Realtime -MaxSamples 1).Value}}
8. Get-VM | Select Name,
   @{N="Provisioned Space (GB)";E={[math]::round($_.ProvisionedSpaceGB,2)}},
   @{N="Used Space (GB)";E={[math]::round($_.UsedSpaceGB,2)}}
9. $vm = Get-VM -Name "VMName"
   $targetHost = Get-VMHost -Name "TargetHost"
   Move-VM -VM $vm -Destination $targetHost
10. $vm = Get-VM -Name "NewVM"
    $spec = New-OSCustomizationSpec -Name "CustomSpec" -Domain "domain.local" -ChangeSid
    Set-VM -VM $vm -OSCustomizationSpec $spec
```

This does not however, make you a programmer ! Yet...

Lab #4.

Write a Complicated PowerShell Program



ChatGPT – PowerCLI Intro

Write a PowerCLI script that looks for all Virtual Machines and Power's on any that are turned off

This is looking more like an App:

Programing is looking now
looking more and more like
knowing how to use ChatGPT

But,.... can we make this more complex..

```
# Connect to vCenter Server
$vcserver = "your_vcenter_server" # Replace with your vCenter Server
$credential = Get-Credential
Connect-VIServer -Server $vcserver -Credential $credential

# Get all VMs
$vm = Get-VM

# Filter VMs that are powered off
$poweredOffVMs = $vm | Where-Object { $_.PowerState -eq "PoweredOff" }

# Power on each VM that is powered off
foreach ($vm in $poweredOffVMs) {
    Write-Host "Powering on VM: $($vm.Name)"
    Start-VM -VM $vm
}

# Disconnect from vCenter Server
Disconnect-VIServer -Confirm:$false
```

ChatGPT – PowerCLI Intro

Write a PowerCLI script that looks for all Virtual Machines and Powers on any that are turned off if CPU utilization is below 70%

Can you make the next program:

1. Error checking ?
2. Time of Day ?
3. Results Table of VMs Turned on
4. Email transaction?
5. Debug output

Next Up: CODE2554LV

Covers how to program with ChatGPT

```
# Connect to the vCenter server
Connect-VIServer -Server "YourVCenterServer"

# Set the CPU threshold percentage
$cpuThreshold = 70

# Get all hosts and their CPU usage
$hosts = Get-VMHost | Select Name, @{Name="CpuUsage";Expression={(($_.CpuUsageMhz / $_.CpuTotalMhz) * 100)}}

# Iterate over each host
foreach ($host in $hosts) {
    # Check if the host's CPU usage is below the threshold
    if ($host.CpuUsage -lt $cpuThreshold) {
        Write-Host "Host $($host.Name) is under the CPU threshold with $($host.CpuUsage)% CPU usage."

        # Get all VMs on this host that are powered off
        $vms = Get-VM -Location $host.Name | Where-Object {($_.PowerState -eq "PoweredOff")}

        # Power on each VM that is powered off
        foreach ($vm in $vms) {
            Write-Host "Powering on VM: $($vm.Name)"
            Start-VM -VM $vm
        }
    } else {
        Write-Host "Host $($host.Name) is over the CPU threshold with $($host.CpuUsage)% CPU usage.
        No VMs will be powered on."
    }
}

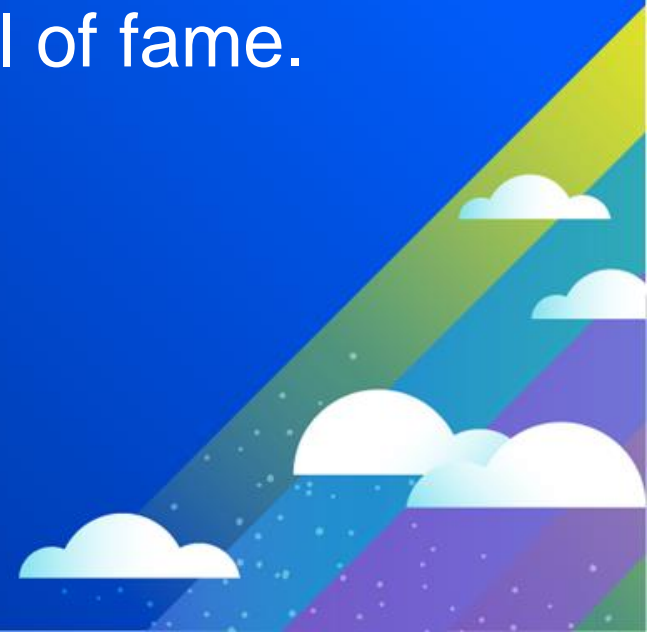
# Disconnect from the vCenter server
Disconnect-VIServer -Confirm:$false
```

Prompt Contest

Submit the most useful prompt fragment to:

eric.nielsen@Broadcom.com

Be sure to include your first and last name for the wall of fame.



vmware® **EXPLORE**

Wall of Fame

Top Submissions!

