StableCode-Instruct-Alpha-3b
API Implementation

Made with <3 by team **Coffee2Code**

Access it on Github & Colab

01

Introducing the project

Project Objective:

Introducing the *StableCode-Instruct-Alpha-3b* API Implementation - a solution that brings the power of AI to projects without local model hosting.

Model Integration:

We harnessed the *StableCode-Instruct-Alpha* 3b model by StabilityAI to craft a functional API. This API leverages the free Colab environment, making it easy for developers to integrate AI-driven features.

Acknowledgment:

A heartfelt acknowledgment to StabilityAI for granting access to the StableCode model. Their support facilitated the creation of a pivotal link between cutting-edge AI and practical implementation.

Purpose:

Create an API using StableCode Instruct Alpha 3b model.

Significance:

Enables users without local hosting capabilities to access the model.

Offers a seamless solution for trying out and implementing the model.

Bridges the gap between advanced AI capabilities and technical limitations.

Accessibility & Ease:

Empowers developers with user-friendly API integration.

Simplifies model utilization, regardless of infrastructure constraints.

Democratizes AI usage by providing an intuitive interface.

02

Getting Started with the project

Getting Started with the project

Google Colab

Local Python Notebook

You can access the model using either our Google colab implementation, or you could either run the model on colab to get the ngrok link, and implement it in your code or project.
You could also use the notebook available on our github repository to run it locally as per availability of resources.

Complete flexibility as per user’s comfort.

03

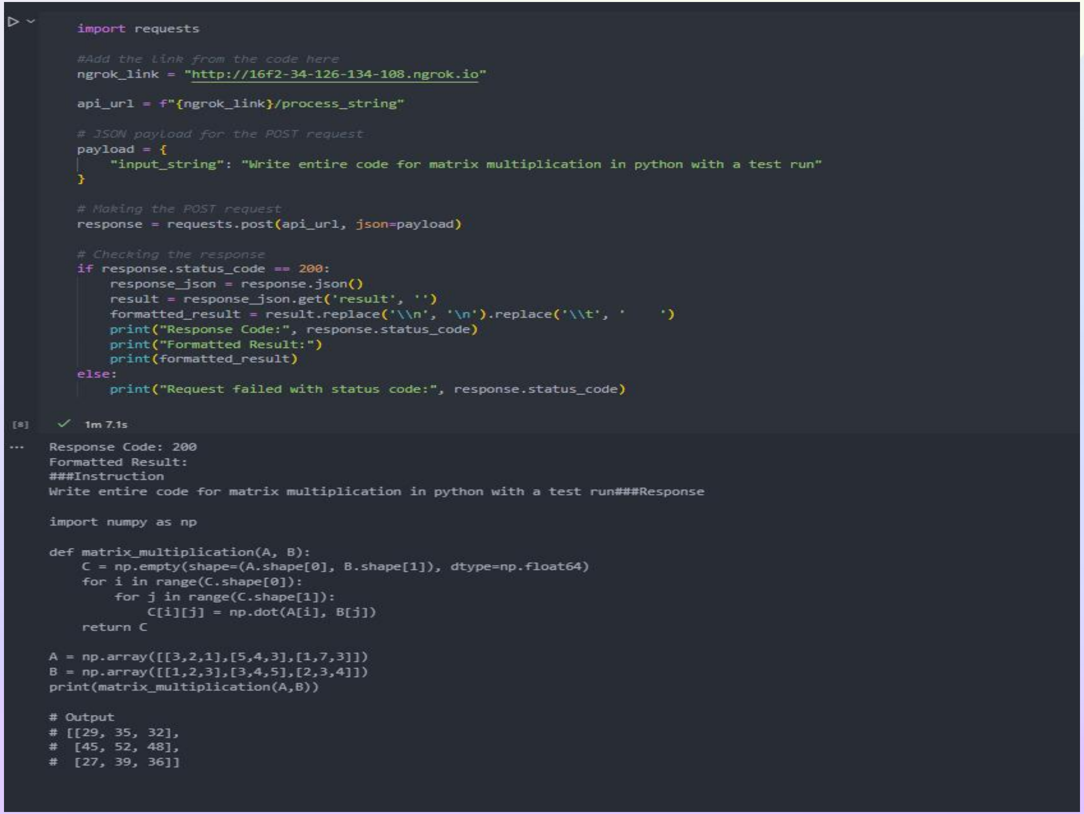
Calling the API in your code

Code to use api in the python code

```
import requests

#Add the link from the code here
ngrok_link = "http://16f2-34-126-134-108.ngrok.io"

api_url = f"{ngrok_link}/process_string"

# JSON payload for the POST request
payload = {
      "input_string": "Write entire code for matrix multiplication in python with a test run"
}

# Making the POST request
response = requests.post(api_url, json=payload)

# Checking the response
if response.status_code == 200:
      response_json = response.json()
      result = response_json.get('result', '')
      formatted_result = result.replace('\\n', '\n').replace('\\t', '')
      print("Response Code:", response.status_code)
      print("Formatted Result:")
      print(formatted_result)
else:
      print("Request failed with status code:", response.status_code)
```

Feel free to copy this sample from our repo

Result using python code



```python
import requests

#Add the link from the code here
ngrok_link = "http://16f2-34-126-134-108.ngrok.io"

api_url = f"{ngrok_link}/process_string"

# JSON payload for the POST request
payload = {
    "input_string": "Write entire code for matrix multiplication in python with a test run"
}

# Making the POST request
response = requests.post(api_url, json=payload)

# Checking the response
if response.status_code == 200:
    response_json = response.json()
    result = response_json.get('result', '')
    formatted_result = result.replace('\\n', '\n').replace('\\t', '    ')
    print("Response Code:", response.status_code)
    print("Formatted Result:")
    print(formatted_result)
else:
    print("Request failed with status code:", response.status_code)
```

[8] ✓ 1m 7.1s

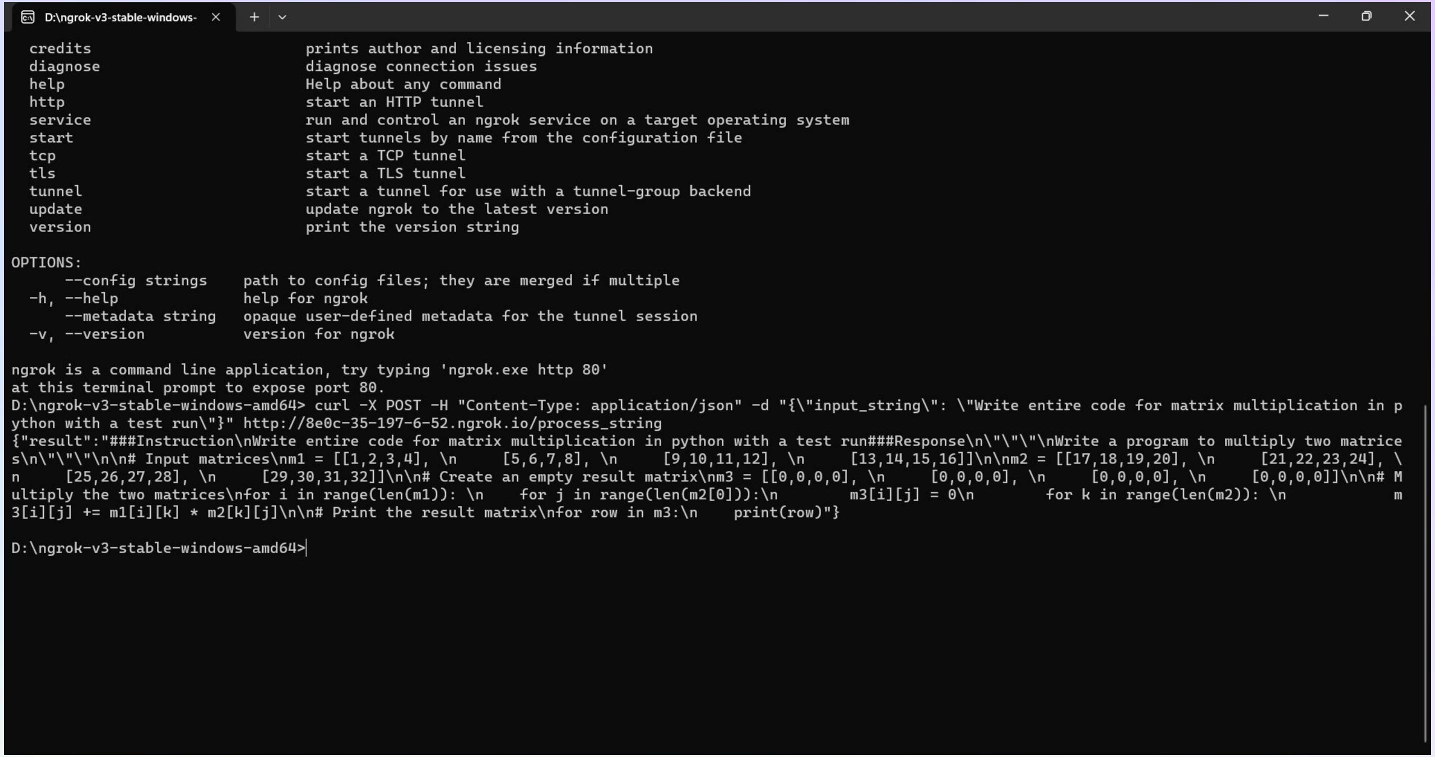
```
Response Code: 200
Formatted Result:
###Instruction
Write entire code for matrix multiplication in python with a test run###Response

import numpy as np

def matrix_multiplication(A, B):
    C = np.empty(shape=(A.shape[0], B.shape[1]), dtype=np.float64)
    for i in range(C.shape[0]):
        for j in range(C.shape[1]):
            C[i][j] = np.dot(A[i], B[j])
    return C

A = np.array([[3,2,1],[5,4,3],[1,7,3]])
B = np.array([[1,2,3],[3,4,5],[2,3,4]])
print(matrix_multiplication(A,B))

# Output
# [[29, 35, 32],
#  [45, 52, 48],
#  [27, 39, 36]]
```

Curl Command

```
curl -X POST -H "Content-Type: application/json" -d "{\"input_string\": \"Write entire code for matrix
multiplication in python with a test run\"}" http://17b7-34-138-136-161.ngrok.io/process_string
```

Feel free to copy this sample from our repo

Result using curl



```
  credits                        prints author and licensing information
  diagnose                       diagnose connection issues
  help                           Help about any command
  http                           start an HTTP tunnel
  service                        run and control an ngrok service on a target operating system
  start                          start tunnels by name from the configuration file
  tcp                            start a TCP tunnel
  tls                            start a TLS tunnel
  tunnel                         start a tunnel for use with a tunnel-group backend
  update                         update ngrok to the latest version
  version                        print the version string

OPTIONS:
      --config strings    path to config files; they are merged if multiple
  -h, --help              help for ngrok
      --metadata string   opaque user-defined metadata for the tunnel session
  -v, --version           version for ngrok

ngrok is a command line application, try typing 'ngrok.exe http 80'
at this terminal prompt to expose port 80.
D:\ngrok-v3-stable-windows-amd64> curl -X POST -H "Content-Type: application/json" -d "{\"input_string\": \"Write entire code for matrix multiplication in p
ython with a test run\"}" http://8e0c-35-197-6-52.ngrok.io/process_string
{"result":"###Instruction\nWrite entire code for matrix multiplication in python with a test run###Response\n\"\"\"\nWrite a program to multiply two matrice
s\n\"\"\"\n\n# Input matrices\nm1 = [[1,2,3,4], \n     [5,6,7,8], \n     [9,10,11,12], \n     [13,14,15,16]]\n\nm2 = [[17,18,19,20], \n     [21,22,23,24], \
n     [25,26,27,28], \n     [29,30,31,32]]\n\n# Create an empty result matrix\nm3 = [[0,0,0,0], \n     [0,0,0,0], \n     [0,0,0,0], \n     [0,0,0,0]]\n\n# M
ultiply the two matrices\nfor i in range(len(m1)): \n    for j in range(len(m2[0])):\n        m3[i][j] = 0\n        for k in range(len(m2)): \n            m
3[i][j] += m1[i][k] * m2[k][j]\n\n# Print the result matrix\nfor row in m3:\n    print(row)"}

D:\ngrok-v3-stable-windows-amd64>
```

Where do we go from here?

Future Vision - VS Code Extension, as free alternative to GitHub Copilot.

Also just as the hackathon started we got the announcement to Meta’s Code Llama, so we could expect an option to implement that too.

Inspiration: We envision extending the functionality of the StableCode Instruct Alpha 3b model to serve as a powerful VS Code extension. Becoming a free open source alternative for the popular copilot by github.

Community Collaboration: We aim to collaborate with the coding community to refine and enhance the extension, making it a versatile and valuable tool for all.

Acknowledgments

- **LabLab.ai:** Our sincere thanks to LabLab.ai for hosting the StableCode 24-hours Hackathon. Your platform provided us with the opportunity to explore, innovate, and bring our ideas to life.
- **StabilityAI:** We deeply appreciate StabilityAI for providing the StableCode-Instruct Alpha 3b model. This cutting-edge model served as the foundation of our project, enabling us to create a functional API with far-reaching potential.
- **Organizers, Mentors, and Contributors:** A special shoutout to all those who played a pivotal role in organizing, mentoring, and contributing to this hackathon. Your guidance, expertise, and encouragement were instrumental in shaping our project.

Thank You!