

Implementing Web Application Integrate Nutrition API

Team ID	PNT2022TMID04940
Project Name	Nutrition Assistant Application

Clarifai:

The screenshot shows the 'Account Settings' page in the Clarifai Community web application. The browser address bar shows 'clarifai.com/settings'. The user is logged in as 'Nithisha S' (NS). The left sidebar contains links for 'Account', 'Billing', 'Security', and 'Usage'. The main content area is titled 'Account Settings' and contains two sections: 'Contact Information' and 'Profile'. The 'Contact Information' section has fields for 'First Name' (Nithisha), 'Last Name' (S), 'Company' (Dr. Sivanthi Aditanar College Of Engineering), 'Country' (India), 'Job Title' (Student), and 'Job Role' (Other). There is an 'Update Contact Information' button. The 'Profile' section has a 'User-ID' field with the value 'clarifaiforme' and a 'Change User id' button. The bottom of the screen shows a Windows taskbar with the date and time as 8:23 PM on 15-Nov-22.

Clarifai Community

clarifai.com/settings

My Apps Community

Switch to Legacy Portal Create an App

NS

NS Nithisha S

Account

Billing

Security

Usage

Account Settings

Contact Information

First Name * Last Name *

Nithisha S

Company * Country *

Dr. Sivanthi Aditanar College Of Engineering India

Job Title * Job Role *

Student Other

Update Contact Information

Profile

User-ID

clarifaiforme Change User id

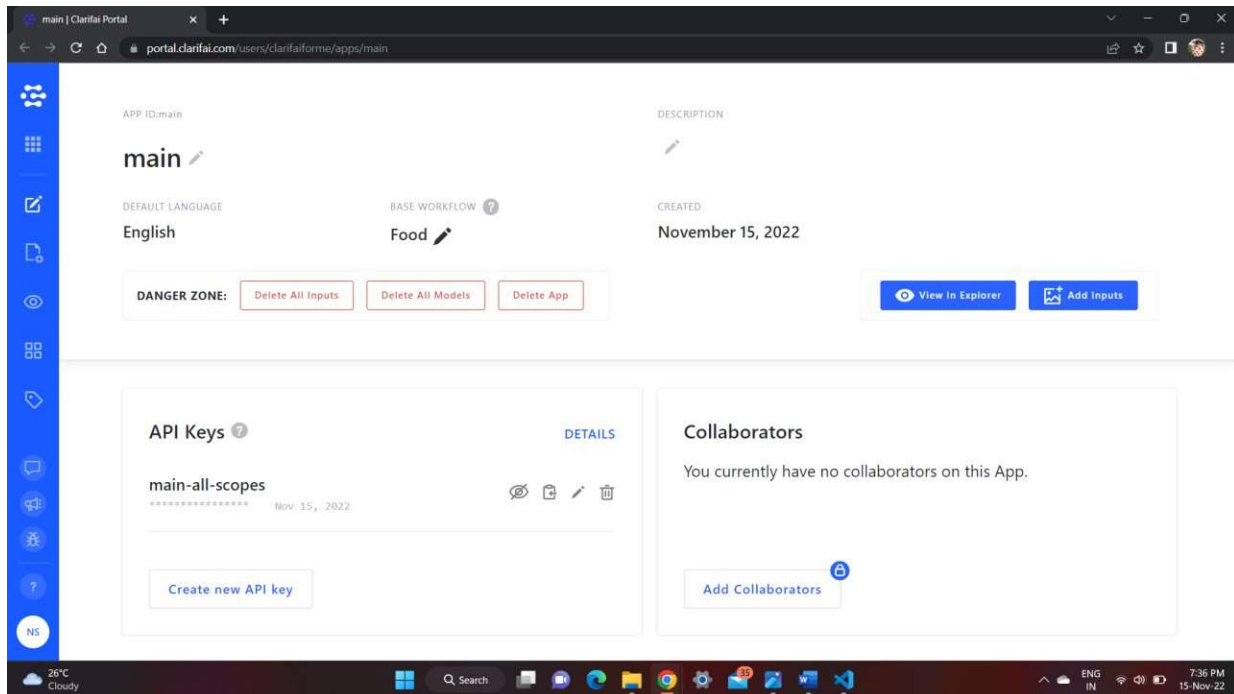
26°C Cloudy

Search

ENG IN

8:23 PM 15-Nov-22

Clarifai API:



Python code to integrate with the Nutrition API:

```
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import resources_pb2, service_pb2, service_pb2_grpc
from clarifai_grpc.grpc.api.status import status_pb2, status_code_pb2
import os

channel = ClarifaiChannel.get_grpc_channel()
stub = service_pb2_grpc.V2Stub(channel)

metadata = (('authorization', 'Key ' + ' '),)

userDataObject = resources_pb2.UserAppIDSet(user_id='clarifaiforme',
app_id='main')

def food_identifier(string):
    with open(string, "rb") as f:
        file_bytes = f.read()

    post_model_outputs_response = stub.PostModelOutputs(
        service_pb2.PostModelOutputsRequest(
            user_app_id=userDataObject,
            model_id="food-item-recognition",
```

```

version_id="1d5fd481e0cf4826aa72ec3ff049e044
inputs=[
    resources_pb2.Input(
        data=resources_pb2.Data(
            image=resources_pb2.Image(
                base64=file_bytes
            )
        )
    )
]
),
metadata=metadata
)

if post_model_outputs_response.status.code != status_code_pb2.SUCCESS:
    print("There was an error with your request!")
    print("\tCode:
{}".format(post_model_outputs_response.outputs[0].status.code))
    print("\tDescription:
{}".format(post_model_outputs_response.outputs[0].status.description))
    print("\tDetails:
{}".format(post_model_outputs_response.outputs[0].status.details))
    raise Exception("Post model outputs failed, status: " +
post_model_outputs_response.status.description)

output = post_model_outputs_response.outputs[0]

print("Predicted concepts:")
for concept in output.data.concepts:
    print("\t%s %.2f" % (concept.name, concept.value))

return output.data.concepts[0].name

```