Exercise 1 Report

Exchange data using HTTP API

Nguyen Tieu Phuong 20210692

# Code & Screenshots

## a) Send data to Thinkspeak API

### - Via URL-encoded params

import requests

import json

api\_key = "T7H40F0X82VGW7L5"

url = "https://api.thingspeak.com/update"

field1 = 20

field2 = 33

url\_with\_params = f"{url}?api\_key={api\_key}&field1={field1}&field2={field2}"

response = requests.post(url\_with\_params)

print("Response from URL with params:", response.text)

print("Data sent using URL parameters.")



### - Via json in request body

import requests

url = "https://api.thingspeak.com/update"

api\_key = "T7H40F0X82VGW7L5"

params = {

    "api\_key": api\_key,

    "field1": 20,

    "field2": 30

}

response = requests.get(url, *params*=params)

print(f"Status Code: {response.status\_code}")

print(f"Response from JSON request body: {response.text}")

A screenshot of a computer

Description automatically generated

## b) Get data from Thinkspeak API

import requests

def fetch\_data():

    url = "https://api.thingspeak.com/channels/1529099/feeds.json?results=2"

    response = requests.get(url)

    if response.status\_code == 200:

        data = response.json()

        feeds = data.get('feeds', [])

        for feed in feeds:

            temperature = feed.get('field1')

            humidity = feed.get('field2')

            print(f"Temperature: {temperature}, Humidity: {humidity}")

    else:

        print("Failed to fetch data from ThingSpeak API")

if \_\_name\_\_ == "\_\_main\_\_":

    fetch\_data()

A screen shot of a computer

Description automatically generated

# Test HTTP request by Postman

## Send data in URL-encoded payload:

A screenshot of a computer

Description automatically generated

## Send data in JSON wrapped body:

A screenshot of a computer

Description automatically generated

## Fetch data from channel:

A screenshot of a computer

Description automatically generated