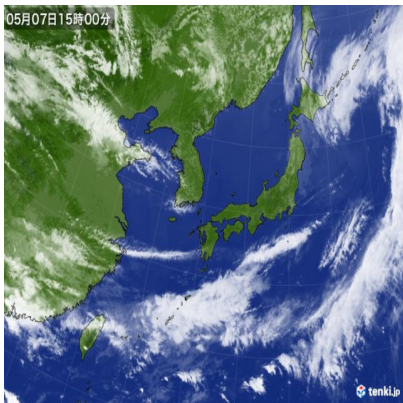




Webã?µã?¤ã??ã?ã??æ°?è±¡ç?»å?•ã??å?ã¾¼ã?ã??ã?ã?ã?ã?ã?ã?ã?ç°¡å??

Description

é?²ã•®æ§?å•ã??å•?å³⁄₄ã•?ã???ã•ã•?ã•ã??ã•?•SDRã??è³¹⁄₄å ¥ã•?ã?¢ã?³ã??ã??ã??ã½?ã??ã•ã•?ã•£ã•
?æ??é??ã•è|ã??ã•³⁄₄ã•?ã??ã??Webã?µã?•ã??ã•?ã??æ??æ??°ã•®èj?æ??ç?»å•?ã??ã??ã?|ã?³ã?-
ã?¹⁄₄ã??ã•?ã??ã•ã•?ã•§ã•?ã??

ã??ã?ã?°ã?©ã? ã•-ã?ã??ã•æ??ã•?ã??

```
import requests
from bs4 import BeautifulSoup
import time
import os

# JMAの天気予報を取得するURL
JMA_URL = "https://tenki.jp/satellite/japan-near/"

# 保存ディレクトリを作成
SAVE_DIR = "./himawari_images"
os.makedirs(SAVE_DIR, exist_ok=True)

def get_latest_image_url():
    try:
        # 天気予報ページを取得
        response = requests.get(JMA_URL, verify=False)
        response.raise_for_status()
        soup = BeautifulSoup(response.text, "html.parser")

        # HTMLから最新の衛星画像のURLを取得
        # 画像のURLは、HTMLの<img>タグのsrc属性から取得
        image_url = None
        elems = soup.find_all("img")
        for img in elems:
            if img.get("id") == "satellite-image":
                image_url = img["src"]
                break
    except Exception as e:
        print(f"Error: {e}")

# 最新の衛星画像のURLを取得
latest_image_url = get_latest_image_url()

# 画像をダウンロード
if latest_image_url:
    response = requests.get(latest_image_url, verify=False)
    response.raise_for_status()
    image_data = response.content

    # 画像を保存
    save_filename = f"{time.strftime('%Y%m%d%H%M%S')}.jpg"
    save_path = os.path.join(SAVE_DIR, save_filename)
    with open(save_path, "wb") as f:
        f.write(image_data)
```

```
if image_url:
    # ç?_â¼ã??ã?¹ã??çµ¶â¼URLã•«â?æ•?
    if image_url.startswith("//"):
        image_url = "https:" + image_url
    return image_url
else:
    print("ç?»â?•URLã??â?•â¼?ã•§ã••ã•¼ã•?ã??ã•§ã•?ã•?ã??")
    return None
except Exception as e:
    print(f"â?¨â?©â?¼: {e}")
    return None

def download_image(url, filename):
    try:
        response = requests.get(url)
        response.raise_for_status()
        with open(filename, "wb") as f:
            f.write(response.content)
        print(f"ç?»â?•ã??ä¿•â¿ã•?ã•¼ã•?ã?: {filename}")
    except Exception as e:
        print(f"ç?»â?•ã??ã?|ã?³ã?ã?¼ã??ã?¨â?©â?¼: {e}")

def main():
    while True:
        image_url = get_latest_image_url()
        if image_url:
            # ç•¼â?¨æ?¥æ??ã??ã??ã?jã?ªã?«â••ã•«â•«ã?•ã??
            timestamp = time.strftime("%Y%m%d_%H%M%S")
            filename = os.path.join(SAVE_DIR, f"himawari_{timestamp}.jpg")
            download_image(image_url, filename)
        else:
            print("ç?»â?•â?•â¼?ã•«â±æ??ã?•ã•¼ã•?ã•?ã??")
            # 10â??ã?•ã?¨ã•«â@?èj?i¼?é•©â@?èªæ?´i¼?
            time.sleep(600)

if __name__ == "__main__":
    main()
```

cronã§ç?»â?•â?•â¼?ã??ã?¹ã?±ã?_ã?¥ã?¼ã?«â??ã?ã??ã?•â?•â¼?ã?ã?ç?»â?•
ã??Discordã?³ã??ã?¥ã??ã??ã?£ã?¼ã?µã?¼ã?•ã?¼ã?«é?•ä¿jã?ã??ã?ªã?©ã?•ã?ã?¨_?æ??é??ã?ã?
?ã??ã?ã?ã??ã?ª??ã?«â½¿ã?ã??ã??ã?®ã?«ã?ã??ã?ã?ã?ã?§ã?ã?ã?•â??ãºç??ã?«ã?¨ã?ã?¼ã??é
çç?¹½ã?ã?ªã?ã??

ã?ã?®ç¨?âºjã?®ã??ã?ã?ã?©ã?ã?ªã??ã?ã?ã??æ?_ã?ã?jã??ã?ã??ã?»ã?©æ??é??ã?ã?ã?ã?ã?ã?ã?
ã?ã?ã?ã?ã?ã?¼ã?«ã?«LLMã«ã?³ã?¼ã??ç??æ?•ã?ã?jã??ã??ã?£ã?jâºã?æ??ç?´ã?ã?ã??ã?ã?ã?
?æ?¹æ³?ã?ªã??æ?°â??ã?§ã??ã?ã?ã?®ã?ã?¼ã?ã??ã??ã??ã?æ?•ã?ã?¼ã?ã??

ã?jã?ªã?¿ã?«ã?jã??ã?£ã?¨èª¿ã?ª¹jã?¿ã?ã?ã?ã?ã?ã?âºã?®æ??ç¨¿ã?®æ?¹æ³?i¼?SDR + Raspberry Pi
NOAAi¼?ã?§ã?¨æ?¥æ?¬ã?®æ°?è±¿jè?æ??ã?ã?¼ã?ã??ã?®ç?»â?•ã?¨ã??ã?ã?ã?ã?ã?ã?ã?ã?
ã??NOAAã??Meteor-Mã?ªã?©ã?¨ã?¨é??ä¿jæ?¹â¼ã?é•ã?£ã?jã?â±æ??ã?ã??ã?
ªã?•ã?¹ã??ã?ã??æ§ç¨?ã?ã?ªã?ã?ã?ã?ã?ã?ã?ã?ã?ã?ã?ã?ã?ã?ã?ã?ã?ã?ã?

Category

- æ?¥ã? ã•®æ´»å??
- ç?-ã??è´´?

Tags

- AI
- LLM
- Python
- Raspberry NOAA
- Raspberry Pi
- SDR
- ä,?é??å??å,?
- åª©æ°?
- å±±æ¢´´ç??

Date Created
2025å'5æ??7æ?¥
Author
kazuo-tsubaki

default watermark