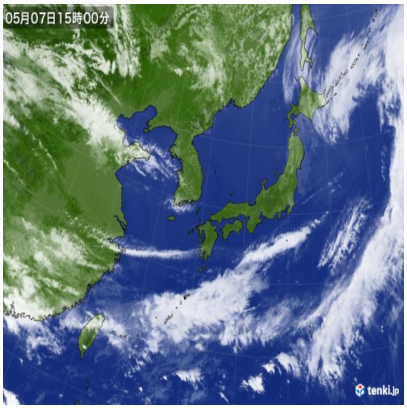

Webã?µã?¤ã??ã•?ã??æ°?è±¡ç?»å?•ã??å•?å¾?ã•?ã??ã• ã•?ã•ªã??ç°¡å•?

Description

default watermark

é?²ã•®æ§?å•ã??å•?å¾?ã•?ã??ã• ã•?ã•ªã??ã?•SDRã??è³¼å ¥ã•?ã?¢ã?³ã??ã??ã??ä½?ã??ã•¨ã•?ã•£ã•?æ??é??ã•¯è¦ã??ã•¾ã•?ã??ã??Webã?µã?¤ã??ã•?ã??æ??æ?°ã•®è¡?æ??ç?»å?•ã??ã??ã?¦ã?³ã?-ã?¼ã??ã•?ã??ã• ã•?ã•§ã•?ã??

ã??ã?ã?°ã?©ã? ã•¯ã?ã??ã•ªæ??ã•?ã??

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

# JMAã•®ã•²ã•¾ã?•ã??ç?»å?•ã??ã?¼ã?¸URLï¼?ä¾?ï¼?æ??æ?°ã•®ç?»å?•ã??å•?å¾?ï¼?
JMA_URL = "https://tenki.jp/satellite/japan-near/"

# ä¿•å?å??ã??ã?£ã?¬ã?¯ã??ã?ª
SAVE_DIR = "./himawari_images"
os.makedirs(SAVE_DIR, exist_ok=True)

def get_latest_image_url():
    try:
        # JMAã•®ã??ã?¼ã?¸ã??å•?å¾?
        response = requests.get(JMA_URL, verify=False)
        response.raise_for_status()
        soup = BeautifulSoup(response.text, "html.parser")

        # ç?»å?•URLã??æ¤?ç´¢ï¼?ä¾?ï¼?HTMLå??ã•®imgã?¿ã?°ã•®srcå±?æ?§ï¼?
        # å®?é??ã•®HTMLæ§?é? ã•«å•?ã?•ã•?ã•¦èª¿æ?´ã•?å¿?è¦•ã•§ã•?
        image_url = None
        elems = soup.find_all("img")
        for img in elems:
            if img.get("id") == "satellite-image":
                image_url = img["src"]
                break
```

```
        if image_url:
            # ç?¸å¯¾ã??ã?¹ã??çµ¶å¯¾URLã•«å¤?æ•?
            if image_url.startswith("//"):
                image_url = "https:" + image_url
            return image_url
        else:
            print("ç?»å?•URLã??å•?å¾?ã•§ã••ã•¾ã•?ã??ã•§ã•?ã•?ã??")
            return None
    except Exception as e:
        print(f"ã?¨ã?©ã?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?»å?•ã??ä¿•å?ã•?ã•¾ã•?ã•?: {filename}")
    except Exception as e:
        print(f"ç?»å?•ã??ã?¦ã?³ã?ã?¼ã??ã?¨ã?©ã?¼: {e}")

def main():
    while True:
        image_url = get_latest_image_url()
        if image_url:
            # ç•¾å?¨æ?¥æ??ã??ã??ã?¡ã?¤ã?«å••ã•«å•«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?»å?•å•?å¾?ã•«å¤±æ??ã•?ã•¾ã•?ã•?ã??")
        # 10å??ã•?ã•¨ã•«å®?è¡?ï¼?é•©å®?èª¿æ?´ï¼?
        time.sleep(600)

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

cronã•§ç?»å?•å•?å¾?ã??ã?¹ã?±ã?¸ã?¥ã?¼ã?«å??ã•?ã??ã?•å•?å¾?ã•?ã•?ç?»å?•ã??Discordã?³ã??ã?¥ã??ã??ã?£ã?¼ã?µã?¼ã?•ã?¼ã•«é?•ä¿¡ã•?ã??ã•ªã•©ã?•ã•?ã•¨ä¸?æ??é??ã•?ã•?ã??ã•°ã••ã??ã•ªã??ã•«ä½¿ã•?ã??ã??ã•®ã•«ã•ªã??ã••ã•?ã•§ã•?ã•?ã?•å??ä°ºç??ã•«ã•¯ã?ã•¾ã??é•¢ç?½ã••ã•ªã•?ã??

ã•?ã•®ç¨?å°¦ã•®ã??ã?ã?°ã?©ã? ã•ªã??ä¸?ã•?ã??æ?¸ã•?ã•¦ã??ã••ã??ã•»ã•©æ??é??ã•?ã•?ã•?ã??ã•ªã•?ã•?ã?•ã?ã?¼ã?«ã?«LLMã•«ã?³ã?¼ã??ç??æ?•ã•?ã•¦ã??ã?ã??ã•£ã•¦å°?ã•?æ??ç?´ã•?ã•?ã??ã•¨ã•?ã•?æ?¹æ³?ã•ªã??æ?°å??ã•§å??ã••ã??ã•®ã•?å¾?ã??ã?ã??ã•¨æ?•ã•?ã•¾ã•?ã??

ã•¡ã•ªã•¿ã•«ã•¡ã??ã•£ã•¨è¹¿ã•¹ã•¦ã•¿ã•?ã•¨ã•?ã?•ã?•å?•ã•®æ??ç¨¿ã•®æ?¹æ³?ï¼?SDR + Raspberry Pi NOAAï¼?ã•§ã•¯æ?¥æ?¬ã®æ°?è±¡è¡?æ??ã•²ã•¾ã?•ã?ã•®ç?»å?•ã•¯å?ã?ã•ªã•?ã?ã•?ã•§ã•?ã??NOAAã??Meteor-Mã•ªã•©ã•¨ã•¯é??ä¿¡æ?¹å¼•ã•?é•?ã•£ã•¦ã?•å¤§æ??ã•?ã?ã•ªã?·ã?¹ã??ã? ã??æ§?ç¯?ã?ã•ªã•?ã•¨ã•?ã•?ã•ªã•?ã?ã•?ã•§ã•?ã??

default watermark

Category

1. æ?¥ã? ã•®æ´»å??
2. ç?¬ã??è¨?

Tags

1. AI
2. LLM
3. Python
4. Raspberry NOAA
5. Raspberry Pi
6. SDR
7. ä¸?é??å??å¸?
8. å¤©æ°?
9. å±±æ¢¨ç??

Date Created
2025å¹´5æ??7æ?¥
Author
kazuo-tsubaki

default watermark