from flask import Flask, request, abort

from linebot import LineBotApi, WebhookHandler

from linebot.exceptions import InvalidSignatureError

from linebot.models import MessageEvent, TextMessage, TextSendMessage

from apscheduler.schedulers.background import BackgroundScheduler

from datetime import datetime, timedelta

import os, re

app = Flask(\_\_name\_\_)

line\_bot\_api = LineBotApi(os.getenv("LINE\_CHANNEL\_ACCESS\_TOKEN"))

handler = WebhookHandler(os.getenv("LINE\_CHANNEL\_SECRET"))

scheduler = BackgroundScheduler()

scheduler.start()

@app.route("/callback", methods=['POST'])

def callback():

signature = request.headers['X-Line-Signature']

body = request.get\_data(as\_text=True)

try:

handler.handle(body, signature)

except InvalidSignatureError:

abort(400)

return 'OK'

@handler.add(MessageEvent, message=TextMessage)

def handle\_message(event):

text = event.message.text

match = re.match(r"リマインド\s\*(.+?)\s(.+)", text)

if match:

time\_str = match.group(1)

content = match.group(2)

remind\_time = parse\_time(time\_str)

if remind\_time:

user\_id = event.source.user\_id

scheduler.add\_job(

func=send\_reminder,

trigger='date',

run\_date=remind\_time,

args=[user\_id, content]

)

line\_bot\_api.reply\_message(

event.reply\_token,

TextSendMessage(text=f"{remind\_time.strftime('%Y/%m/%d %H:%M')} にリマインドします！")

)

else:

line\_bot\_api.reply\_message(

event.reply\_token,

TextSendMessage(text="日時の形式が分かりませんでした。例: '明日10時 メール送信'")

)

def send\_reminder(user\_id, content):

line\_bot\_api.push\_message(

user\_id,

TextSendMessage(text=f"🔔 リマインド：{content}")

)

def parse\_time(time\_str):

now = datetime.now()

if "明日" in time\_str:

hour\_match = re.search(r"(\d{1,2})時", time\_str)

if hour\_match:

hour = int(hour\_match.group(1))

return now.replace(day=now.day+1, hour=hour, minute=0, second=0, microsecond=0)

return None

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

app.run()