

Site Checker with Time Scheduling

This Python application periodically attempts to connect to a specified website or server according to a defined schedule, such as every few minutes or at specific times. It continuously monitors the availability of the site.

Key Features:

1. **Automated Monitoring:** This Python script automates the process of checking the availability of a specified website (URL) at regular intervals (CHECK_INTERVAL minutes).
2. **Notification Mechanisms:**
 - **Email Alerts:** Utilizes smtplib to send email notifications to NOTIFY_EMAIL when the website is found to be down.
 - **Console Feedback:** Provides real-time feedback in the console, indicating whether the site is up or down and its HTTP status code.
3. **Robust Error Handling:** Handles potential connection errors (requests.ConnectionError) gracefully and notifies the user appropriately.
4. **Configurability:** Users can easily configure the script by adjusting URL, CHECK_INTERVAL, EMAIL_ADDRESS, EMAIL_PASSWORD, and NOTIFY_EMAIL to suit their monitoring needs.
5. **Continuous Operation:** Utilizes schedule and time.sleep(1) to ensure continuous execution, running the website check function (check_website) as per the defined interval.

How to Use:

1. **Setup and Configuration:**
 - Replace URL with the website or server you want to monitor.
 - Adjust CHECK_INTERVAL to specify how often (in minutes) the script should check the website's status.
 - Configure EMAIL_ADDRESS, EMAIL_PASSWORD, and NOTIFY_EMAIL for email notifications.
2. **Email Configuration:**
 - Ensure the SMTP server settings (smtp.gmail.com, port 587 with TLS encryption) are correct for your email provider.
 - Update EMAIL_ADDRESS and EMAIL_PASSWORD with your email credentials.
3. **Execution:**
 - Run the script (python your_script_name.py) to start monitoring the specified URL.
 - The script will continuously check the website's status at the defined intervals and notify via email if the site is down.
4. **Deployment Considerations:**
 - Deploy the script on a server or a machine that runs continuously (e.g., cloud server, Raspberry Pi) to ensure uninterrupted monitoring.
 - Monitor console outputs for real-time updates on the website's status.

