Website Availability monitoring using Python Modules

Welcome! to the documentation of website availability monitoring using python modules.

Objective:

Our objective is to create a python script using python modules to monitor a website availability and get the status of the website.

1.Checking the website availability:

Modules used to check the website availability, status of the website and saving the result in a log file.

- Urllib package
- Request
- Logging
- Requests

Urllib:

Urllib package is the **URL handling module for python**. It is used to fetch URLs (Uniform Resource Locators). It uses the urlopen function and is able to fetch URLs using a variety of different protocols.

Urllib is a package that collects several modules for working with URLs, such as:

- urllib.request for opening and reading.
- urllib.parse for parsing URLs
- urllib.error for the exceptions raised
- urllib.robotparser for parsing robot.txt files

If urllib is not present in your environment, execute the below code to install it.

pip install urllib

urllib.request:

syntax: ***from urllib import request***

This module helps to define functions and classes to open URLs (mostly HTTP). One of the most simple ways to open such URLs is:

urllib.request.urlopen(url)

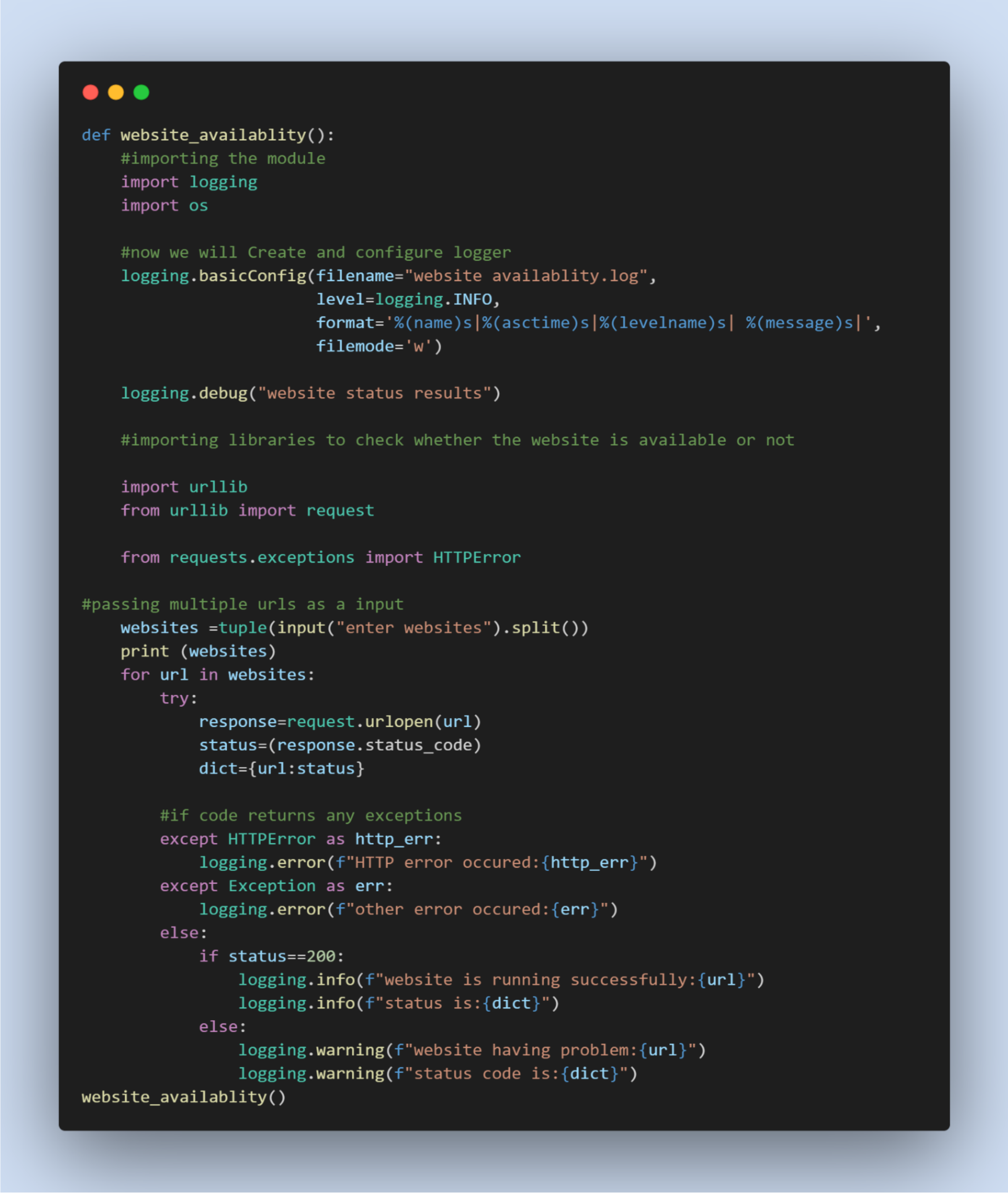
requests:

Requests library is one of the integral part of Python for making HTTP requests to a specified URL. The requests module **allows you to send HTTP requests using Python**. The HTTP request returns a Response Object with all the response data (content, encoding, status, etc).One of the most common HTTP methods is GET. The GET method indicates that you’re trying to get or retrieve data from a specified resource. To make a GET request,

requests.get(url).

For response *response=requests.get(url)*

Program:

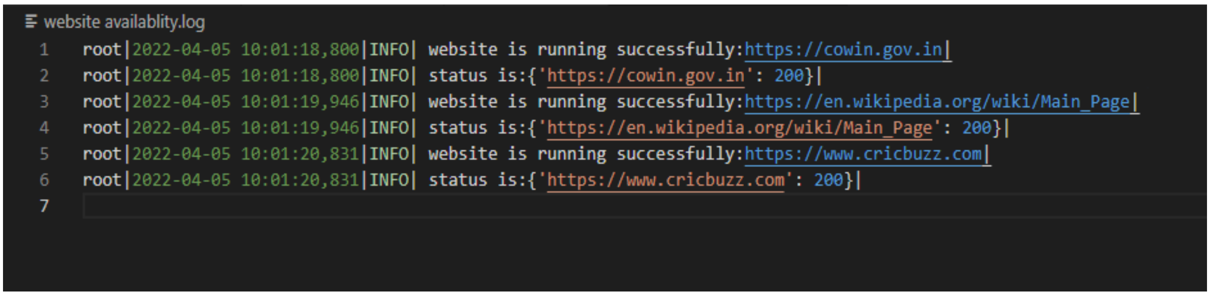


```
def website_availablity():
    #importing the module
    import logging
    import os

    #now we will Create and configure logger
    logging.basicConfig(filename="website availablity.log",
                        level=logging.INFO,
                        format='%(name)s|%(asctime)s|%(levelname)s| %(message)s|',
                        filemode='w')

    logging.debug("website status results")

    #importing libraries to check whether the website is available or not

    import urllib
    from urllib import request

    from requests.exceptions import HTTPError

#passing multiple urls as a input
    websites =tuple(input("enter websites").split())
    print (websites)
    for url in websites:
        try:
            response=request.urlopen(url)
            status=(response.status_code)
            dict={url:status}

        #if code returns any exceptions
        except HTTPError as http_err:
            logging.error(f"HTTP error occured:{http_err}")
        except Exception as err:
            logging.error(f"other error occured:{err}")
        else:
            if status==200:
                logging.info(f"website is running successfully:{url}")
                logging.info(f"status is:{dict}")
            else:
                logging.warning(f"website having problem:{url}")
                logging.warning(f"status code is:{dict}")
website_availablity()
```

Logging module:

We used logging module to record or track the events while we run the code or program and saving the record in a log file for future use.

Output:



```
root|2022-04-05 10:01:18,800|INFO| website is running successfully:https://cowin.gov.in|
root|2022-04-05 10:01:18,800|INFO| status is:{'https://cowin.gov.in': 200}|
root|2022-04-05 10:01:19,946|INFO| website is running successfully:https://en.wikipedia.org/wiki/Main_Page|
root|2022-04-05 10:01:19,946|INFO| status is:{'https://en.wikipedia.org/wiki/Main_Page': 200}|
root|2022-04-05 10:01:20,831|INFO| website is running successfully:https://www.cricbuzz.com|
root|2022-04-05 10:01:20,831|INFO| status is:{'https://www.cricbuzz.com': 200}|
```

2.Sending email:

after checking the website availability we are sending the result in a mail with the log file.

Modules used:

- smtplib
- schedule
- time

smtplib:

Python, being a powerful language don’t need any external library to import and offers a native library to send emails- “SMTP lib”. “smtplib” creates a Simple Mail Transfer Protocol client session object which is used to send emails to any valid email id on the internet. In this article, we are using a Gmail account to send a mail. Port number used here is ‘587’. And if you want to send mail using a website other than Gmail, you need to get the corresponding information. different websites using different port numbers.

1. First of all, “smtplib” library needs to be imported.
2. After that, to create a session, we will be using its instance SMTP to encapsulate an SMTP connection.
   s = smtplib.SMTP('smtp.gmail.com', 587)

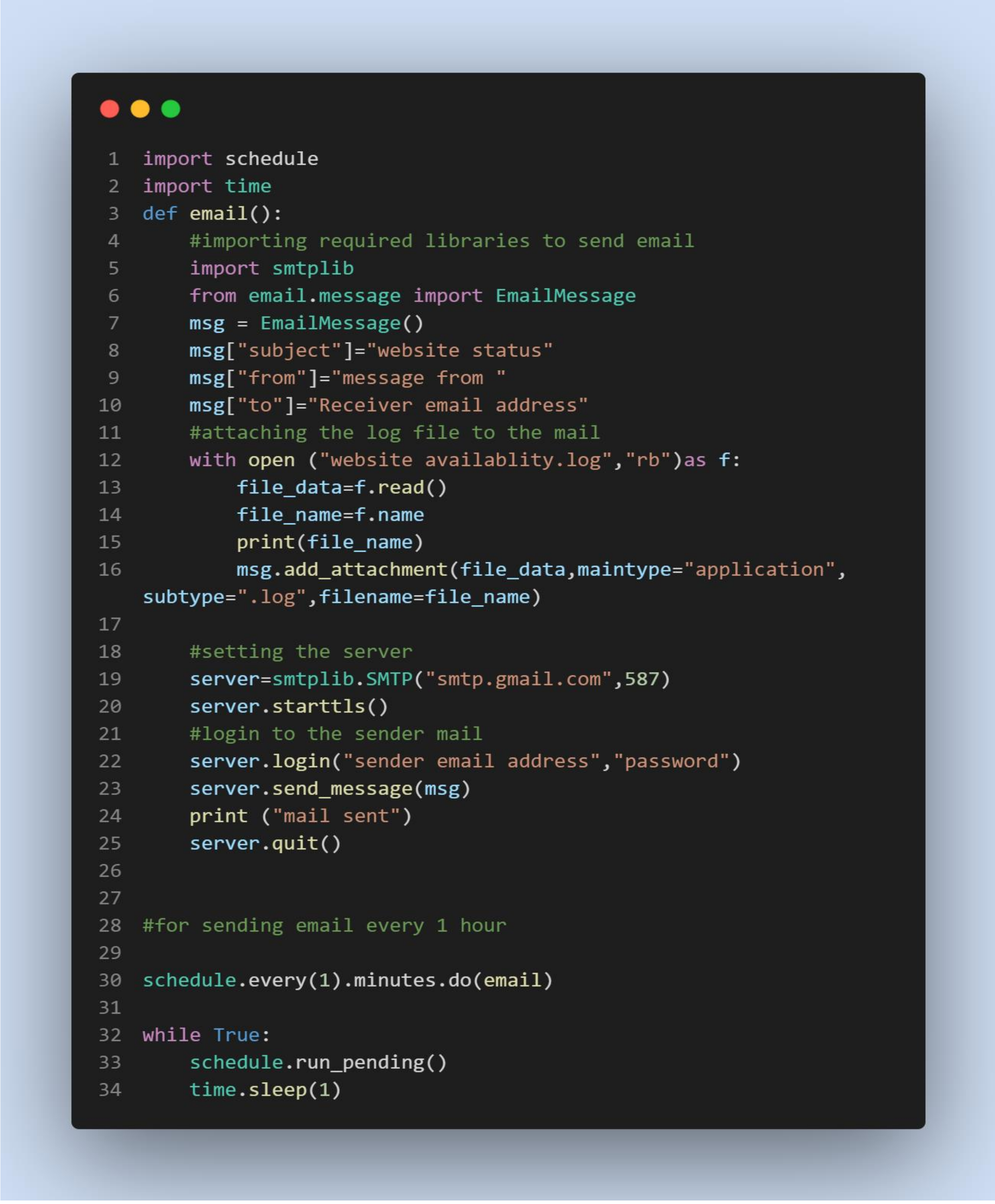
   In this, you need to pass the first parameter of the server location and the second parameter of the port to use. For Gmail, we use port number 587.

3. For security reasons, now put the SMTP connection in the TLS mode. TLS (Transport Layer Security) encrypts all the SMTP commands. After that, for security and authentication, you need to pass your Gmail account credentials in the login instance.The compiler will show an authentication error if you enter invalid email id or password.

4. Store the message you need to send in a variable say, message. Using the sendmail() instance, send your message. sendmail() uses three parameters: **sender_email_id, receiver_email_id and message_to_be_sent**. The parameters need to be in the same sequence.

This will send the email from your account. After you have completed your task, terminate the SMTP session by using quit().

Program:



```
import schedule
import time
def email():
    #importing required libraries to send email
    import smtplib
    from email.message import EmailMessage
    msg = EmailMessage()
    msg["subject"]="website status"
    msg["from"]="message from "
    msg["to"]="Receiver email address"
    #attaching the log file to the mail
    with open ("website availablity.log","rb")as f:
        file_data=f.read()
        file_name=f.name
        print(file_name)
        msg.add_attachment(file_data,maintype="application",
subtype=".log",filename=file_name)

    #setting the server
    server=smtplib.SMTP("smtp.gmail.com",587)
    server.starttls()
    #login to the sender mail
    server.login("sender email address","password")
    server.send_message(msg)
    print ("mail sent")
    server.quit()


#for sending email every 1 hour

schedule.every(1).minutes.do(email)

while True:
    schedule.run_pending()
    time.sleep(1)
```

SCHEDULE:

For scheduling the mail, we will make use of the **schedule** package in python. It is very lightweight and easy to use.

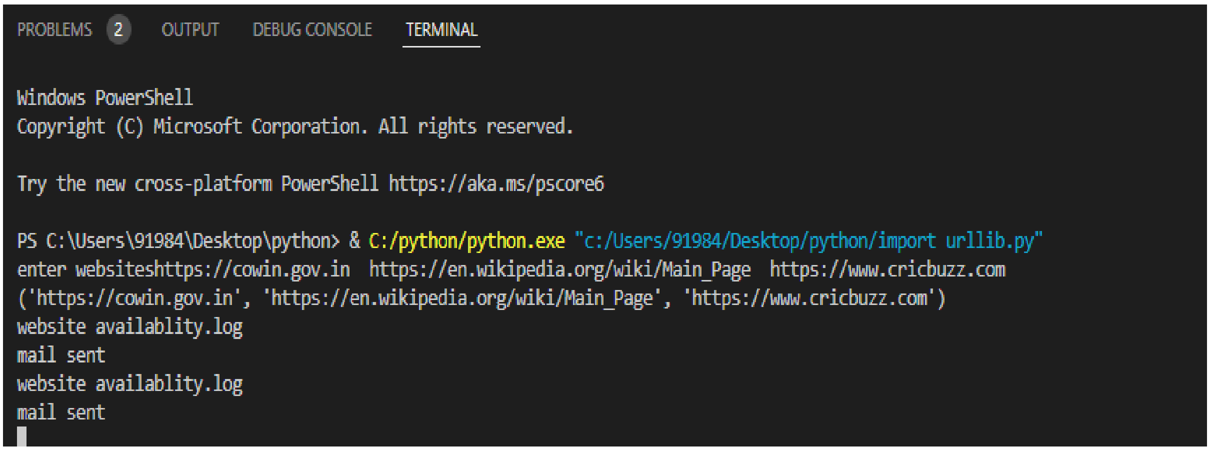
Install the module

pip install schedule

The below function will call the function mail every 2 seconds.

schedule.every(2).seconds.do(mail)

output:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\91984\Desktop\python> & C:/python/python.exe "c:/Users/91984/Desktop/python/import urllib.py"
enter websiteshttps://cowin.gov.in  https://en.wikipedia.org/wiki/Main_Page  https://www.cricbuzz.com
('https://cowin.gov.in', 'https://en.wikipedia.org/wiki/Main_Page', 'https://www.cricbuzz.com')
website availablity.log
mail sent
website availablity.log
mail sent
```

Conclusion:

A python script for monitoring website availability and sending a alert mail is created successfully