

Python Module to Monitor Website Availability

What is Website Monitoring?

Website Monitoring is an all-encompassing term for any activity that involves testing a website or web service for availability, performance, or function. A Website Monitoring service checks and verifies that the site is up and working and site visitors can use the site as expected.

The term “Website Monitoring” refers to any activity that checks the availability, performance, and function of a website or web service. Typically, the term refers to automated testing or Real User Monitoring, but some sites still do not test at all or rely on periodic checks performed by employees. Manual testing is sporadic and undependable when considering the number of variables that influence a site’s availability, performance, and function.

How does Website Monitoring work?

Automated website Monitoring uses a network of computers located near the site’s end users. This network of computer checkpoints interacts with a website or service to verify that the service works as expected. The monitoring system designates a checkpoint to test the site, and the checkpoint may go through several steps to conduct the test. The checkpoint:

1. Initiates a connection with the website or service
2. Checks the return for a response code. For basic availability, the checkpoint reports the result and considers the test complete, but for more advanced monitoring the checkpoint continues.
3. Checks the return for specified content
4. Loads the content into a real browser (Real Browser Monitoring)

5. Records load times for each page element as it loads in the browser
(performance monitoring)
6. Attempts to log in, conduct a search, use a shopping cart, even complete a purchase (web-application monitoring)
7. Reports its findings back to the monitoring service

If the result includes errors or slow response times, the service may initiate the check again from a different checkpoint to verify a persistent error before alerting the website's support team.

Monitoring may also use data coming from the actual users of the website or service. Known as Real User Monitoring (RUM), script files, agents, cookies, or server-side code track the website's performance as each site visitor accesses the site. RUM provides real user insights, but RUM relies on user interaction with the website to get data making it an unviable method for tracking uptime.

What types of Website Monitoring are there?

Website Monitoring involves testing websites for availability, performance, and function and alerting support staff when the page doesn't work as expected. Typically, a monitor type will fall into one of the previously mentioned categories, although the more advanced monitors may cover all three.

Availability monitoring

Availability is about uptime, or in other words, making sure a website or service is always accessible and to some degree functional. Availability can involve web services, domains, and pages.

Basic website and API monitoring – These basic monitors check for a successful response or a specific response from websites and APIs that support HTTP protocol, and they may perform basic authentication. Basic availability monitors can also

measure the timing and size of the response and issue alerts for slow response times. HTTP(S) monitors do not load the content into a browser, but the monitoring service may check the response for the presence or absence of specified words, phrases, or a regular expression.

Server availability – As long as a server or device recognizes TCP/IP protocol, a monitoring service can verify availability for the device and selected ports. A monitoring service can check the availability as frequently as once per minute preventing costly downtime and lost productivity over the web or behind the firewall.

Advanced Availability – These specialized automated monitors verify DNS records, check for proper configuration of SSL certificates, query databases, log into email servers, and download files from FTP servers.

Performance monitoring

Performance monitoring checks a website's or service's speed. Performance monitors track the time for connection speeds (frontend and backend) and browser load times. Performance monitors may utilize Synthetic Monitoring or RUM technology. RUM and the Full Page Check provide the most comprehensive performance data set. The Full Page Check gives detailed performance data for every element on the page. Performance monitors issue alerts for page errors, missing content, and slow performance.

Functionality monitoring

Web Application Monitors or transaction monitors test a site's functionality. These specialized monitors use script files that interact with forms, site search, shopping carts, and payment systems. Transaction monitors interact with a web application in the same way as regular visitors, and they typically verify the predictable "happy

paths” for completing a task. If an error occurs or the performance drops, the system issues an alert to the support staff. Many different errors can block users from using a web application that availability and performance monitors cannot catch.

PROGRAM:

```
1  #Importing libraries
2  import time
3  import hashlib
4  from urllib.request import urlopen, Request
5
6  # setting the URL you want to monitor
7  url = Request('https://google.com/',
8  |             | headers={'User-Agent': 'Mozilla/5.0'})
9
10 # to perform a GET request and load the
11 # content of the website and store it in a var
12 response = urlopen(url).read()
13
14 # to create the initial hash
15 currentHash = hashlib.sha224(response).hexdigest()
16 print("running")
17 time.sleep(10)
18 while True:
19     try:
20         # perform the get request and store it in a var
21         response = urlopen(url).read()
22
23         # create a hash
24         currentHash = hashlib.sha224(response).hexdigest()
25
26     ⚡ # wait for 30 seconds|
27     time.sleep(30)
28
29     # perform the get request
30     response = urlopen(url).read()
31
32     # create a new hash
33     newHash = hashlib.sha224(response).hexdigest()
34
35     # check if new hash is same as the previous hash
36     if newHash == currentHash:
37         continue
```

```

38
39     # if something changed in the hashes
40     else:
41         # notify
42         print("something changed")
43
44         # again read the website
45         response = urlopen(url).read()
46
47         # create a hash
48         currentHash = hashlib.sha224(response).hexdigest()
49
50         # wait for 30 seconds
51         time.sleep(30)
52         continue
53
54     # To handle exceptions
55     except Exception as e:
56         print("error")

```

Output:

```

viswanath@Viswanath-PC:~$ /bin/python3 "/home/viswanath/Desktop/Monitoring Website/main.py"
running
something changed
error
error
error
error
error
error
error
error
error
error
^Z
[1]+  Stopped                  /bin/python3 "/home/viswanath/Desktop/Monitoring Website/main.py"
viswanath@Viswanath-PC:~$

```

CONCLUSION:

If the website's availability, performance, and function are mission critical for a business or an organization, Website Monitoring is essential. For most businesses, a properly functioning website is important whether the site provides marketing collateral, e-commerce, or a service. The competition awaits just a few clicks away, and frustrated end users will abandon a down, poorly performing, or malfunctioning site and do business with the competing site instead.