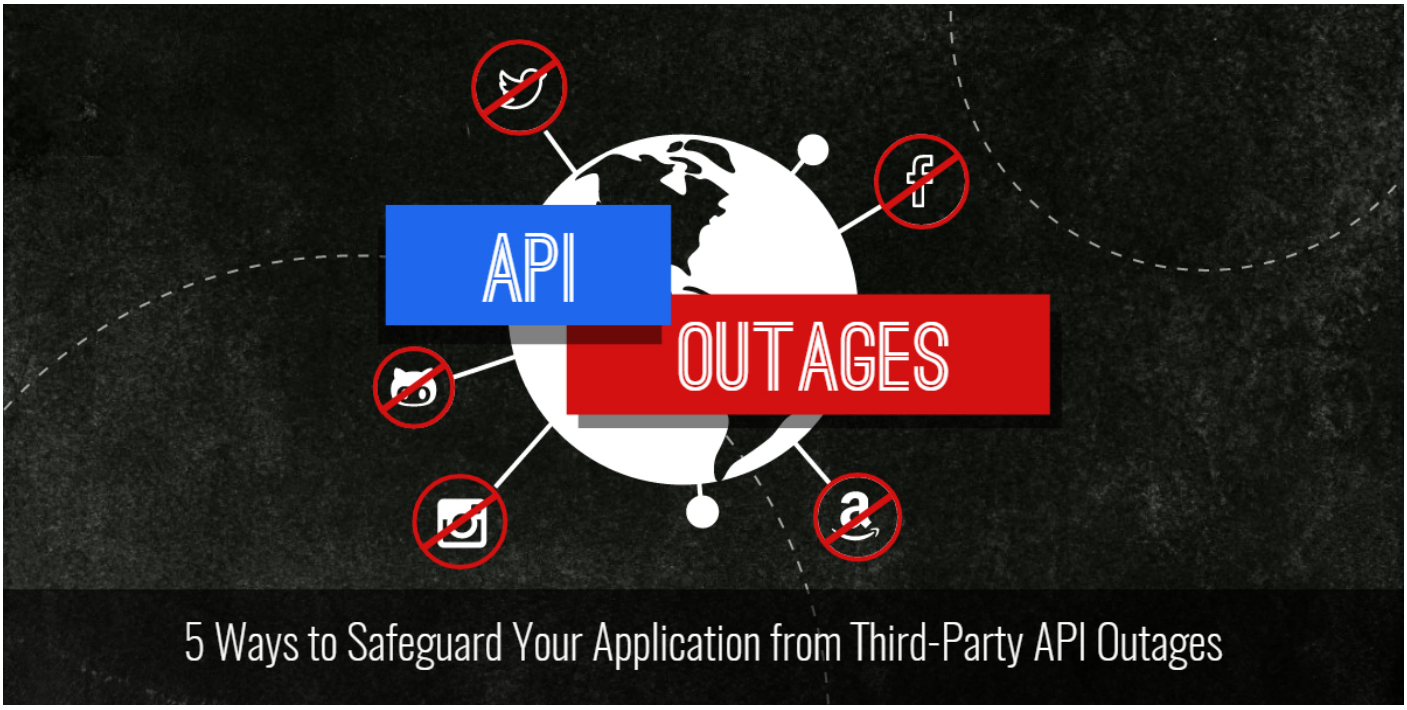
5 Ways to Safeguard Your Application from Third-Party API Outages



The Internet of Things (IoT) is growing at a rapid pace with millions of new devices getting connected every day.

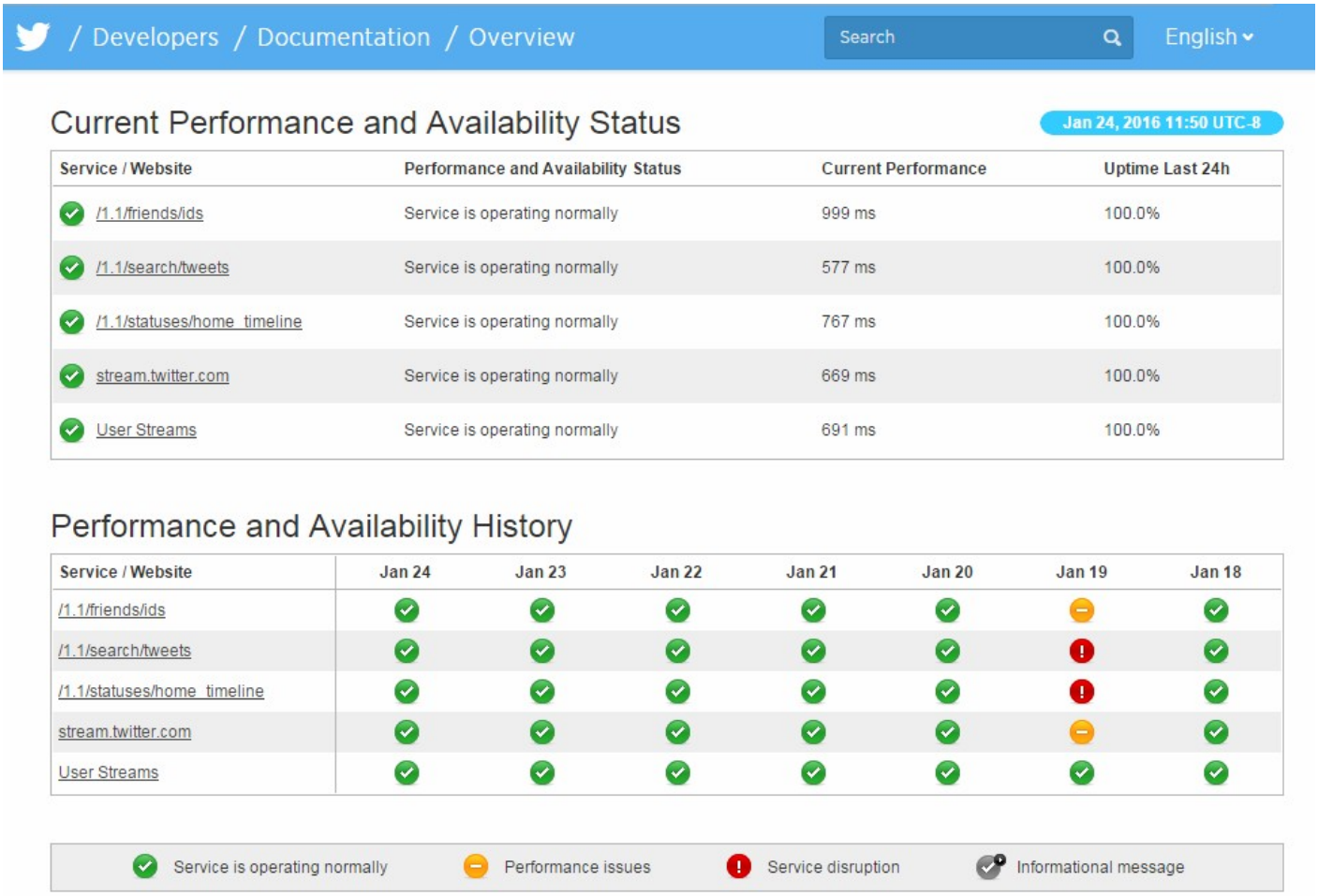
Gartner forecasts that 6.4 billion connected things will be in use in 2016 and by 2020, the number will reach 20.8 billion. Billions of devices including smartphones, laptops, cars, watches, refrigerators, and even thermostats are all getting connected thanks to APIs. Many of these devices are part of a larger application ecosystem, connected to the cloud and each other using the same APIs.

There have been a number of high-profile API outages in recent months that caused some well-known websites and applications to go down. With

so many devices and applications using many of the same cloud services and APIs, one outage can potentially take down entire application ecosystems. While it may not be possible to completely prevent an API outage from taking down your website, web or mobile application, it is possible to take steps to reduce the impact of a third-party API outage.

Web Service Outages

Twitter experienced a major outage for about an hour on January 19, 2016 which caused the Twitter website and applications to malfunction for users worldwide.



Application ecosystems are getting larger and more complex with each passing day. While API providers do their best to ensure API availability,

outages do still occur from time to time.

Consider the following examples:

- On January 26, 2015, both the Facebook and Instagram API servers went down for about an hour taking down Facebook and Instagram. The outage also impacted a number of well-known websites including Tinder and HipChat.
- On September 20, 2015, Amazon Web Services (AWS) experienced a brief disruption that caused an increase in faults for the EC2 Auto Scaling APIs. Within the same month, the Facebook Graph API went down briefly on three separate occasions. Facebook is considered a very reliable service so the fact that Facebook suffered from three outages in the same month was widely reported.
- On January 19, 2016, Twitter experienced a major outage that impacted many of the websites and applications using Twitter APIs. The API outages caused the Twitter website and applications to malfunction for users worldwide for about an hour.

API outages aren't limited to the large scale APIs like those experienced by giants like Facebook and Amazon. And if you have an application that is dependent on external APIs to perform a critical function, you need to have a plan for dealing with disruptions.

There are a number of solutions that developers can use to prevent API outages from negatively impacting a website or application.

1. API Virtualization

API virtualization is a process that makes it possible to create a virtual copy of an API capable of emulating the same functionality and performance of a

production API. API virtualization can be used to create a virtual sandbox for integration, scenario, load, and performance testing of internal and third-party APIs. Unlike service virtualization, a process that mimics an entire system, API virtualization only emulates the API which is considerably less expensive, easier to implement, and less time consuming.

"It's all about time and cost. API virtualization, beyond simple service mocking, provides technical teams a frictionless way to reduce delays in development and testing. Lightweight, virtual APIs empower teams to simulate errors, network latency and congestion, and even stand up API sandboxes," said Paul Bruce, API team product manager at SmartBear. "These 'virts' (as the folks at SmartBear call them), allow you overcome third-party downtime, safeguard you from running over subscription limits, and even help to simplify versioning and release logistics."

Using API virtualization tools like ServiceV Pro, developers can test third-party API integrations by mimicking errors, server, and network behavior.

Matti Hjelm, product owner at SmartBear, said that "ServiceV works from two angles: both from the dev perspective to quickly create APIs for own use or for others to use; and for testers to have something to test against. This will truly speed up the build and deploy workflows dramatically."

API virtualization can help developers anticipate how specific third-party API outages will impact their application allowing them to plan for outages before moving the application to general availability.

2. Synthetic and Real User Monitoring

Synthetic Monitoring

Synthetic monitoring is a script or tool that is capable of monitoring various aspects of APIs, web and mobile applications. Synthetic monitoring effectively “simulates users” monitoring the paths that users may take while using an application. Synthetic monitoring can help developers discover problems with an application before users do. Using synthetic monitoring, developers can find out:

- How well the website/application performs from specific geographic locations.
- How long it takes web pages to load.
- How an outage or slowdown will impact users.
- How well third-party scripts and APIs are operating.
- How well transactions are performing.
- How new features will impact performance and speed.

These are just a few of the things that developers can learn about an application by using synthetic monitoring.

"On the production side, synthetic API monitoring provides critical visibility into how your APIs perform under real conditions, prove uptime and availability, and split out important layers of your user experience so you can involve the right people to quickly resolve problems when they arise," said Bruce.

Real User Monitoring

Real user monitoring (RUM) is a script or tool designed to monitor and analyze all user transactions of a website or application. RUM is based on the traffic of actual application users, not virtual or simulated users like synthetic monitoring. RUM shows how users are interacting with an application, providing metrics that reflect the user experience (UX).

RUM can be used to monitor and capture a wide variety of web and mobile application metrics including load time, speed, errors, transaction performance, availability, and more. Problems with third-party APIs can impact the speed and performance of a website or application. There are RUM tools that can send automatic alerts in the event of a problem such as an API outage or application slowdown.

3. A/B Testing

A/B testing is a method of testing in which two variants of a web page or application are compared to each other. For example, two versions of an application; one that includes a third-party API and one that does not, would be compared to see how that API will impact application speed and performance. A/B testing tools use data and metrics to determine the positive or negative impact of specific changes made to web pages or applications.

4. Asynchronous Scripting

Asynchronous scripting helps web pages render faster by allowing the main body of content to be loaded without having to wait for all of the external scripts to load. Without asynchronous scripting, users must wait for all of the external scripts to load first before they can view the main content of the web page. If a problem with a third-party script or API occurs, asynchronous scripting allows the body of content to be loaded despite the problem.

Many API providers offer APIs via JavaScript SDKs with code that is asynchronous by default. Third-party JavaScript APIs and code snippets that are not already asynchronous can be loaded asynchronously. In

HTML5 for example, external JavaScript files can be loaded asynchronously using the async tag:

<script **async** src="script.js"></script>

5. Caching

Caching API responses allows API data to be stored on your website server instead of relying on the third-party server for every API call. Caching helps prevent an application from being impacted by third-party API outages because data is pulled from a local cache file instead of directly from the API server. Caching scripts are available in PHP, Python, and many other languages.

Conclusion

This article covers only a few of the steps that can be taken to safeguard your application from third-party API outages and errors.

Limiting the number of APIs your application uses, using well-designed APIs from reputable API providers, and using a full-featured APM tool can also help safeguard your application from API outages.

With the number of connected devices and application ecosystems growing at such as rapid pace, it is more important than ever for applications to be safeguarded from third-party API outages and errors.

Safeguard your API from third party outages and other performance setbacks. Find out how SmartBear's Ready! API platform of API testing tools can help.