

# Synthetic Monitoring 101





# Table of contents

Introduction.....	1
1. Implement a well thought-out monitoring strategy.....	2
2. Establish your monitoring strategy goals.....	3
3. Start synthetic monitoring early in the development lifecycle and test your monitoring checks fully before implementing.....	4
4. Assign responsibility.....	6
5. Get strategic with documenting check scenarios.....	6
6. Document how alerts are resolved.....	7
7. Choose the right metrics.....	8
8. Use multiple monitoring locations.....	9
9. Combine functional checks with API-che cks.....	10
10. Last, but not least: create versatile checks.....	10
Conclusion.....	11

# Introduction

Performance monitoring of website and applications is a challenging but critical task designed to help ensure a positive end user experience, and to help companies gain and retain loyal customers.

According to Apica's 2017 Consumer Expectations survey, 60% of consumers indicated they'd be less loyal to brands after poor website and app performance, with more than three quarters of respondents saying they expect sites and apps to perform faster than three years ago.

So to avoid 'Digital Desertion,' brands are turning to the experts when it comes to forming formidable monitoring strategies. In this guide, we can help you to understand how synthetic monitoring can add value. We'll explore strategy and how to start, synthetic monitoring and the development lifecycle, alert ownership and resolution, documenting checks, the benefits of geographically dispersed monitoring, and other specific best practices surrounding synthetic monitoring.

# 1. Implement a well thought-out monitoring strategy

Synthetic Performance Monitoring is a monitoring process initiated proactively by external agents that imitate actual end users and web traffic. All companies are likely to have some idea as to where to best focus monitoring. At first, this may just be a hypothesis, but as you collect more and more data, you'll get closer to building a strong strategy surrounding what to monitor.

In considering your strategy, it's best to start by analyzing your applications in order to identify where you must focus your monitoring. In order to do this, you must:

- Classify your systems based on importance
- Determine the most common/business critical tasks – this is where monitoring starts
- Consider the most relevant scenarios and all the possible user journeys
- Establish everything that needs to be monitored within the website and mobile application (including internal interfaces). This is likely to include processes such as making a transaction and performing searches.

An important factor in a successful monitoring strategy is also to identify and include all major stakeholders in the monitoring strategy development process, including developers, quality assurance, IT, management, owners, and marketing. It's also essential to continue to meet with stakeholders to review the strategy monthly or bi-monthly in order to keep it up to date and continuously improve it.

## 2. Establish your monitoring strategy goals

Synthetic monitoring, at a high level, is valuable for determining the availability, performance, and functionality of applications. When determining the specific monitoring to perform against your applications, it's essential to design the scripts and checks to match those goals.

For instance, availability monitoring is typically much simpler, and can be run at higher frequencies than performance or functionality monitoring, but those advantages come at the cost of detailed information.

For your most critical applications, it makes sense to take a blended approach that uses some high-frequency availability monitoring in combination with lower-frequency (but more complex and detailed) performance and functionality monitoring.

# 3. Start synthetic monitoring early in the development lifecycle - and test your monitoring checks fully before implementing

Unfortunately, monitoring is often an afterthought - something that we discuss or talk about once a website or application has gone into production. This is often due to lack of resources or finances, but more often than not it's simply down to lack of planning.

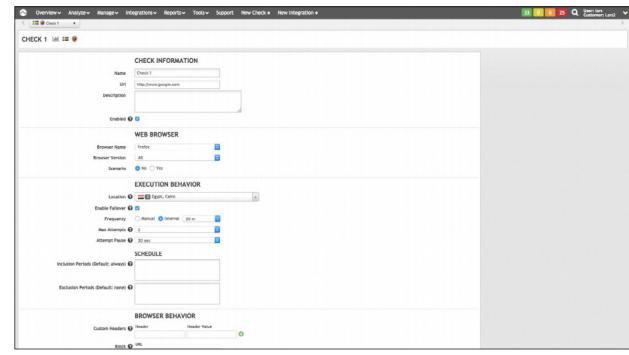
What companies don't often realize is that monitoring should be implemented in the design phase - as you're developing an application or as you're planning and analyzing that application.

Synthetic monitoring is best used before a web application goes into production to monitor its behavior, providing a report on the performance expected in the production stage. These 'pre-production' produced results are a great baseline, and the same synthetic monitoring script can also be continued in the production environment.

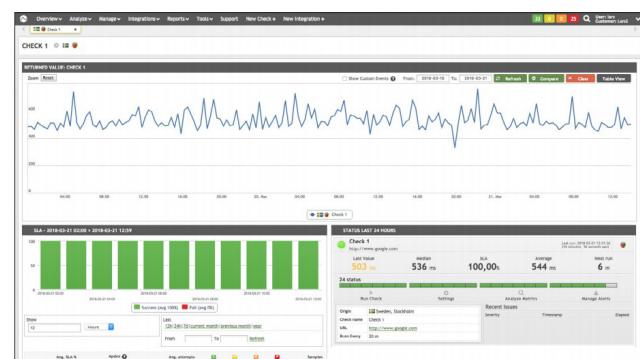
When synthetic monitoring is carried out through design, planning and into production, it acts as a proactive way of identifying issues before they become problems.

## Test your monitoring checks before fully implementing

It may sound obvious, but time needs to be allocated in order to test monitoring checks fully before implementing, not only to ensure they are working, but also to give your team time to get used to them and customize them to suit specific requirements.



When you let new checks run for a reasonable test period, you're in a much better position to establish suitable values for settings such as threshold levels, number of retries and to verify the functionality of the script itself. This way you can customize the check to your liking through trial and error so that it's perfectly suited for your specific needs.



## 4. Assign responsibility

It's essential to identify owners to handle each part of the monitoring and evaluation process to avoid checks breaking when changes are made.

When changes are made to the target environment, you must make sure someone (preferably those who are responsible for the service the checks are connected to) is responsible for each script or check. This is when having different dashboards for different teams and stakeholders really becomes helpful.

These views can help give you a clear overview of relevant data and KPIs, making it easier to keep track of performance at a glance.

## 5. Get strategic with documenting check scenarios

It's common to have complex check scenarios with multiple steps, and most organizations will also want to maintain scripts over an extended period of time. This is when strategic, structured documenting procedures become essential.

You should always document all steps with screenshots, function, input/output, and anything else critical to running checks - effectively, the documentation should explain why the check works the way it does.

Using a naming standard for both groups and checks is also a good idea. This makes it easy for the stakeholders and team members to get a clear overview so that future check maintainers can contribute to your work when you can't.

# 6. Document how alerts are resolved

This is one of the most important best practices surrounding synthetic monitoring. It's essential to plan and configure check alerts that work for your team. And once set up, alerts should only be sent to the relevant party, not to all users.

This is critical when it comes to the alerts being properly addressed and not discarded or ignored - if everyone receives every alert, they may assume someone else is dealing with it.

Specifically, different thresholds should trigger different types of alerts to different people. For example, alerts surrounding minor issues can be sent to a junior engineer, while alerts surrounding more severe issues should be sent to a more senior team member of the team.

The screenshot shows the 'RECIPIENTS' section of the Apica Synthetic Monitoring interface. On the left, under 'USERS', there is a list of recipients for user 'LARS': 'Lars Phone [0799999999]', 'Lars Email [lars.mail@apicasystem.com]', 'Email Custom Message [lars.mail@apicasystem.com]', and 'ServiceNow [ServiceNow]'. Each item has a status indicator (green for Enabled, red for Disabled) and a 'PRIORITY' column. Below this, there is a configuration panel for a Slack integration, including fields for 'Service' (set to 'Slack'), 'Target Name' (empty), 'URL' (set to 'hooks.slack.com/services/'), 'Channel' (empty), 'Message Title' (empty), and 'Message' (empty). A button at the bottom of this panel says '+ Add Slack WebHook Target'. On the right, under 'GROUPS', there is a list with one entry: 'Disabled' (status green). At the top of the page, there is a navigation bar with links like 'Overview', 'Analyze', 'Manage', 'Integrations', 'Reports', 'Tools', 'Support', 'New Check', 'New Integration', and a search bar. The top right corner shows the user 'lars' and the customer 'Lars3'.

*Alerts settings in Apica Synthetic Monitoring*

## 7. Choose the right metrics

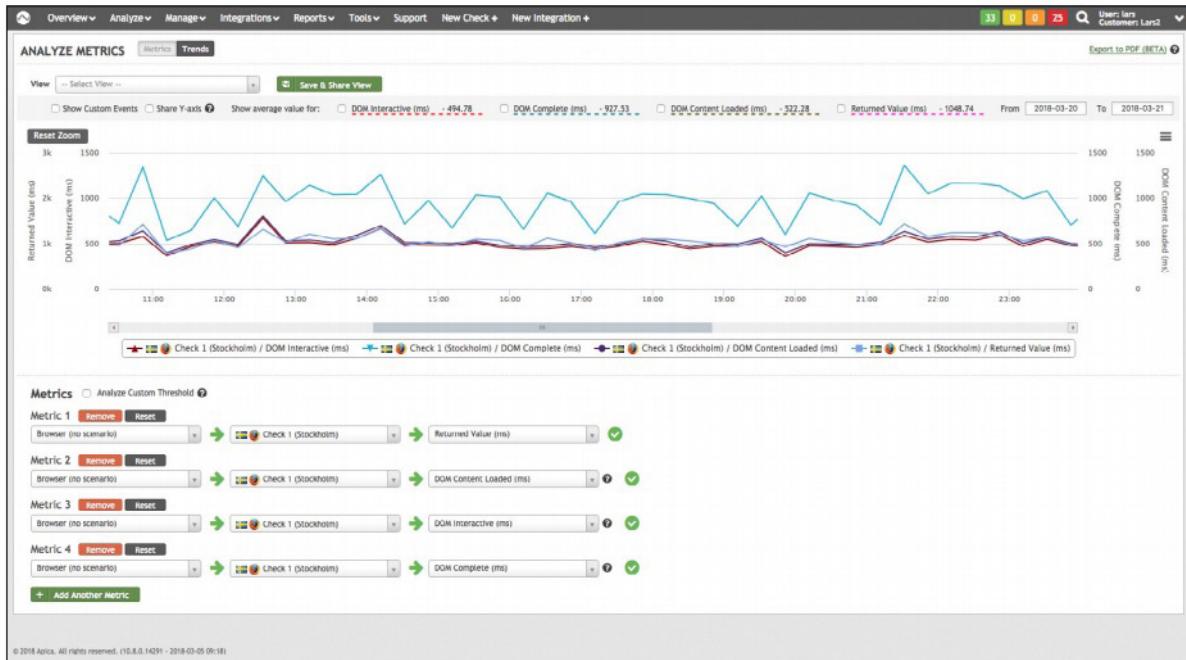
There is no metric that fits all applications, and companies will gradually begin to understand the ones that are best suited to them. It's good practice to compare the measured metrics result with the actual user experience, as they should be similar.

It is important to use the appropriate returned value metrics for the type of application being monitored. Examples of return value metrics include:

- Total browser render time
- DOM complete
- DOM content loaded
- Total download size

To illustrate the use of different metrics, consider a single page application. It would not be measured with the response time metric "DOM complete," as this would show a misleading result and response time due to how the application is built.

Compare the measured metrics result with the actual user experience in order to choose the correct one, they should be similar. There is no metric that fit all applications, and it's up to you to choose the one best suited for yours.



*Different metric options shown in Apica Synthetic Monitoring*

## 8. Use multiple monitoring locations

Develop a location-based monitoring strategy to keep an eye on performance for your audiences in all important regions and help you analyze performance across geographies.

By using a solution with a worldwide network of monitoring probes, you can address business requirements in specific regions or countries where you have users. For example, if your user base is located in Germany and Spain, it's essential that you have check locations in these countries in order to simulate the correct type of traffic there, and to collect accurate data against this simulated traffic.

This is also handy when it comes to ruling out location specific errors, avoid false negatives, and to cover any routing issues over the internet.

If you are doing business across a large number of regions or countries, using multiple monitoring locations can help you analyze and compare performance across these geographies, helping you to understand where to allocate data resources.



## 9. Combine functional checks with API-checks

To really get the full picture of your performance, combine functional checks with APIchecks.

It means you'll get better insight into specific functionalities, giving you the ability to pinpoint issues that may be difficult to spot with only functional or only API monitoring.

It's also a great way to speed up your mean time to recovery, giving you more in-depth knowledge about different functionalities in your application.

## 10. Last, but not least: create versatile checks

Versatility is everything in checks: By creating checks using scripts that are nonspecific, you can easily adapt to use them for multiple scenarios, ultimately saving you time and effort.

In this way, scripts should be easily adapted to changes being made on a specific page so that they don't have to be constantly rewritten. So don't configure tests to always pick the same specific item or product from the same place each time on a webpage - configure them to pick the first product that appears on the page.

# Conclusion

Synthetic monitoring is proving a must-have for companies who know they risk losing customers to less than optimal functionality and performance. From testing new features before deployment, to looking at performance degradations before and after deployments, to testing new markets or geographies, to detecting performance issues related to specific browsers, resolutions and devices - the use cases are endless.

However, while all companies recognize the importance of synthetic monitoring, some may not have the correct solution in place, while others may not understand the importance of underpinning synthetic monitoring with a strong, end-to-end strategy.

Here's a 10 point breakdown of the best practices described in the white paper:

- Implement a well thought-out monitoring strategy
- Establish your monitoring strategy goals
- Start synthetic monitoring early in the development lifecycle & test your monitoring checks fully before implementing
- Assign responsibility
- Get strategic with documenting check scenarios
- Document how alerts are resolved
- Choose the right metrics
- Use multiple monitoring locations
- Combine functional checks with API-checks
- Last, but not least: create versatile checks

## More content on the topic Synthetic Monitoring:

Learn more about the product Apica Synthetic Monitoring and its features

On-demand webinar: How leading fintech innovator Klarna uses Apica synthetic to ensure 99.99% uptime

Watch our 1 minute video to learn more about Apica

# Discover Apica

Leading enterprises rely on the Apica Web Excellence Suite to test and monitor their mission critical business systems, APIs, web and mobile applications. Apica enables businesses to get detailed real-time performance, uptime and capacity insights, ensuring outstanding end user experience and optimized IT operations. Apica's suite - available as SaaS, on-premise and hybrid solutions - is trusted by 300+ leading brands globally. Apica has offices in Stockholm, New York, London and Santa Monica.

To learn more about Apica, visit  
[apicasystems.com](http://apicasystems.com)  
Contact Us: [info@apicasystems.com](mailto:info@apicasystems.com)

Apica Systems

Stockholm  
New York  
London  
Santa Monica

[apicasystems.com](http://apicasystems.com)

