

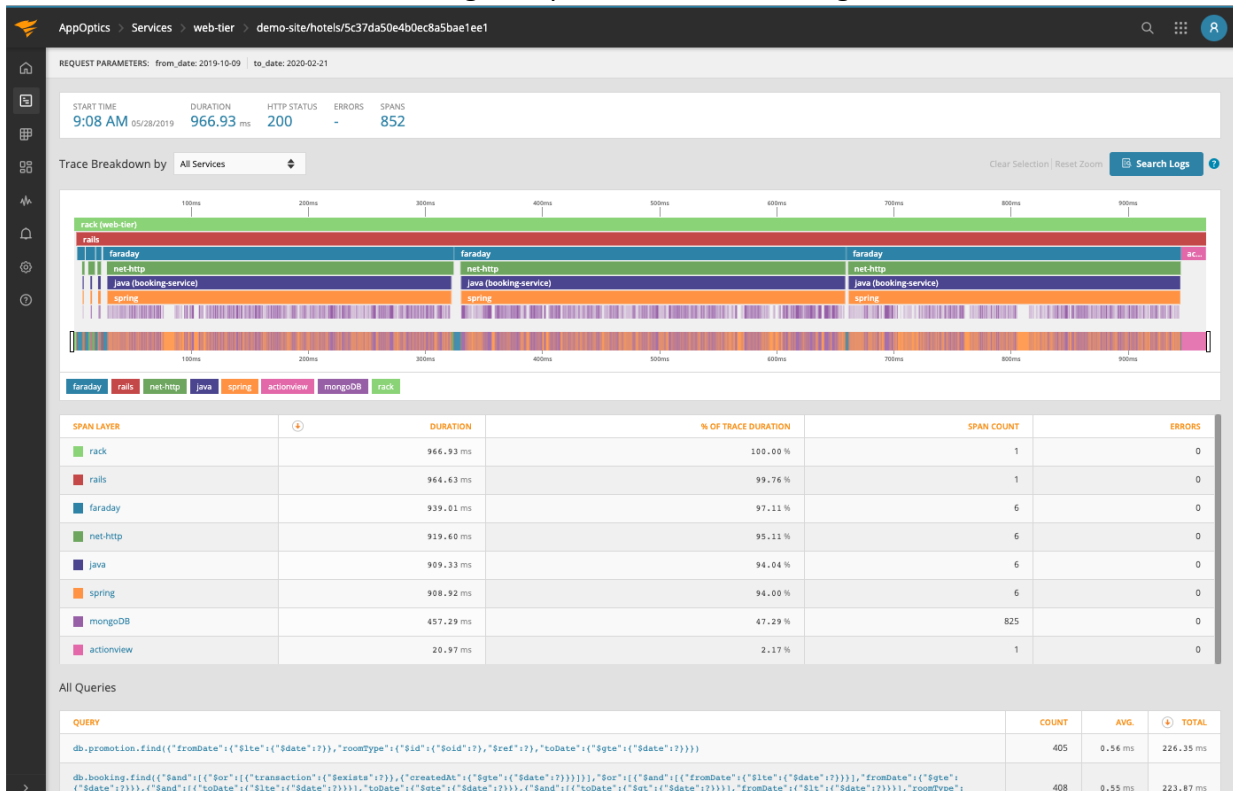
The AppOptics APM Troubleshooting Triple Threat

Distributed trace, live code profiling, and exception tracking

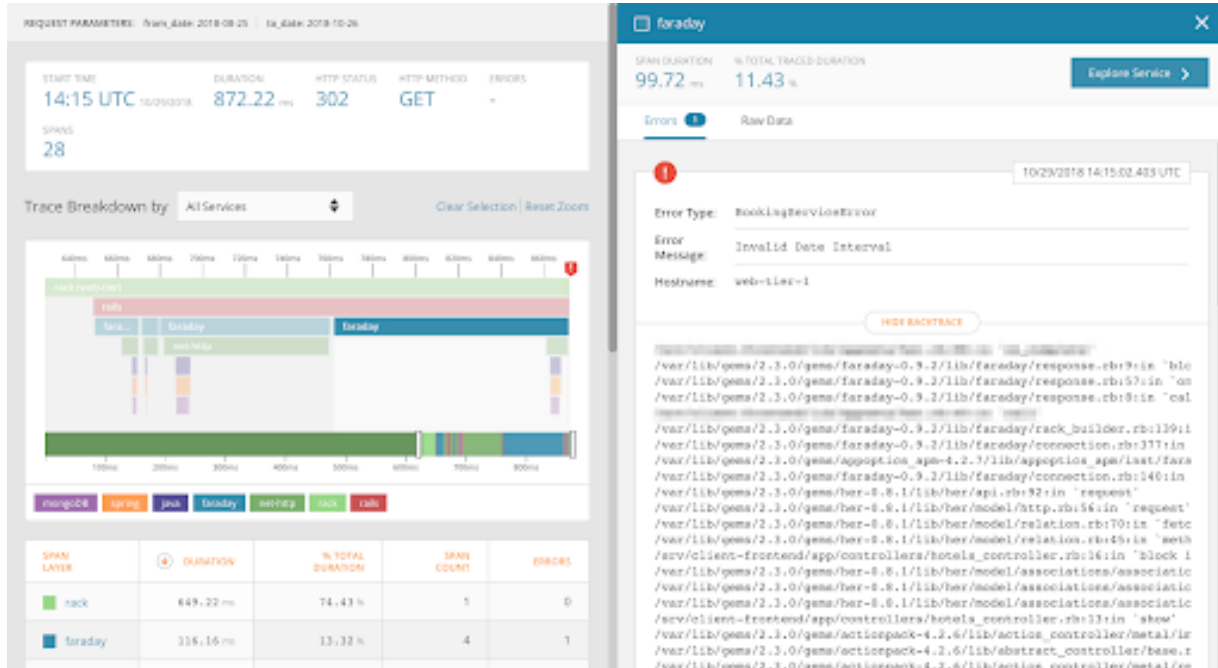
When you think about it, it seems to always come down to time. How much time has the application been down? How long have your users been experiencing a problem? How much time will it take to get your application up and running again? Stuff happens, and when it does, time is of the essence, so you need to get at the root cause fast and minimize the time it takes to get your application back to performing as designed.

The [SolarWinds® AppOptics™](#) product team is always looking at ways to improve, and that has resulted in the triple threat of troubleshooting—three powerful ways AppOptics helps you get to the root cause of performance problems (distributed tracing, live code profiling, and exception tracking). When all three are combined, you’ve got what you need to quickly and accurately identifying what is causing your applications’ poor performance.

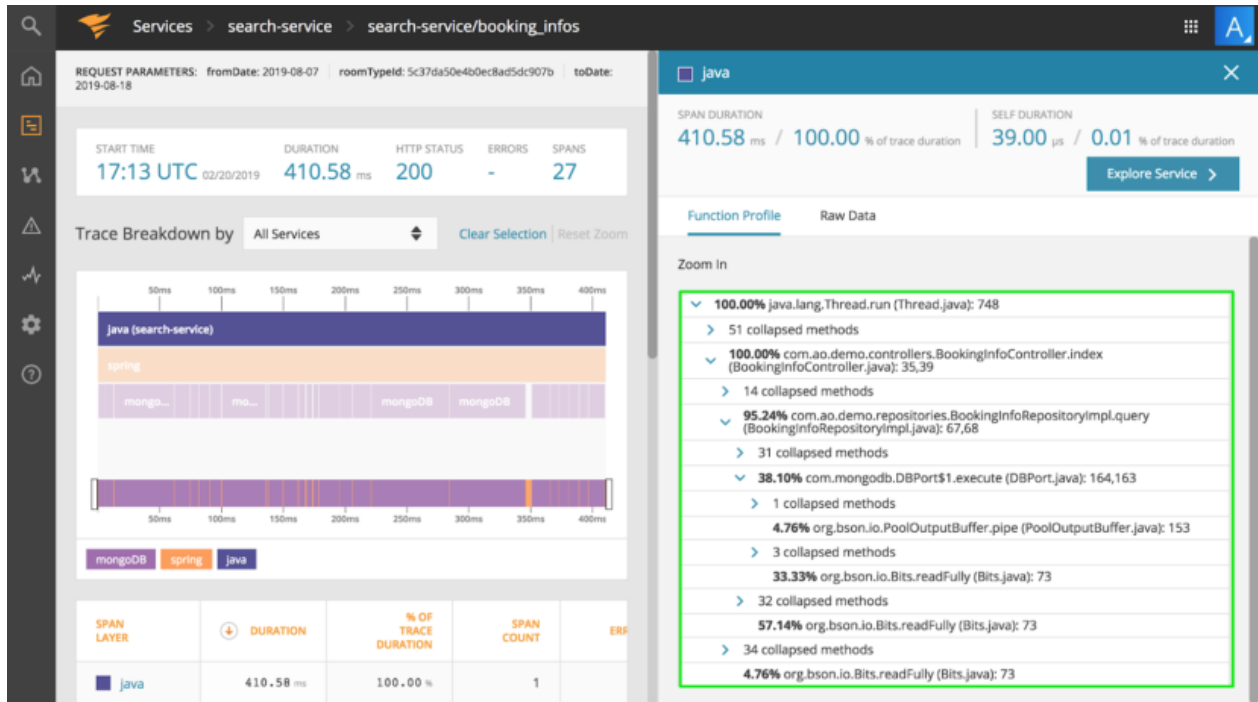
1) Distributed transaction tracing enables you to follow a request as it traverses multiple services and is key to understanding the performance of distributed applications. AppOptics gathers traces continuously and mines them for data; then, you can drill down into each one as shown in the example below. You can get started quickly because AppOptics distributed tracing is auto-instrumented, and no coding is required to take advantage of it.



2) Exception tracking builds on top of the AppOptics automatic tracing capability, providing a summary of exceptions happening within a service context. When troubleshooting a service, click on the Exceptions tab at the top of the page to understand the current volume of exceptions, when they started, and which exceptions are most frequent. You can even drill into each exception, viewing the stack trace, and which endpoints in the application are throwing them.



3) Live code profiling provides a breakdown of the most frequently-called functions and methods in a transaction. Profiling is gathered in a manner tailored to each language, and typically includes information down to the class, method, and even filename and line number. It provides enough detail to understand what line of code is causing a performance issue, and includes the information needed to find the relevant section in the source code quickly.



- 1) Using exception tracking can help you focus on the issues that need attention. For example, if it's 3:00 a.m. and you just got paged, you likely want to know the scope of an issue to understand if it needs to be resolved immediately.
- 2) Now that you've identified it's a problem that needs to be solved, you dive into the trace. AppOptics exception tracking enables you to drill down from a summary of exceptions to view a specific exception in the context of a distributed trace. You can use the trace breakdown to dig deeper into what service was causing the problem.
- 3) Finally, live code profiling will break down these components to identify precisely where the bottleneck is, down to what line of code was causing the performance issue—reducing the amount of troubleshooting and headaches.

There you have it—the AppOptics APM troubleshooting triple threat! Fast, powerful, and automated troubleshooting of application performance problems. The best way to learn more about [SolarWinds AppOptics](#) is to give it a try. Sign up for a [14-day free trial](#).