



# 3 Ways to Improve Your Dev Team Productivity with Rookout

No more frustrated hours spent debugging. Bugs can be solved much more quickly by teams being able to work together. Developers are able to share captured snapshot data with the team to collaborate more efficiently.



# 3 Ways to Improve Your Dev Team Productivity with Rookout

In today's software development ecosystem, developers are the backbone of digital transformation. Organizations are putting a newfound emphasis and focus on maximizing the productivity of these developers. However, maximizing developer productivity is not as simple as it may seem, and many factors can affect it. Yet, in order to achieve it, there is one critical aspect that developers need to have: the proper tools. Having the right tools in place can make developers more productive and even improve overall job satisfaction. Rookout is one of those tools that is a must have in the tool belt of development teams wanting to reduce the time it takes to identify the root cause of issues.

In this blog post, we're going to dive into three ways that you can improve your development team productivity with Rookout. Rookout is a tool that allows you to debug an application in any environment without the need for changing code, redeploying, or needing to have a locally replicated instance of your application. Let's dive into how you can improve your overall development velocity and have some fun doing it with a cool new technology.

## KEY TAKEAWAYS:

It is crucial that developers have the proper tools in order to maximize their productivity. No more frustrated hours spent debugging. Bugs can be solved much more quickly by teams being able to work together.

There are three central ways that development team productivity can be improved with production grade debugging technologies like Rookout.

First, they help teams to improve the mean time to repair (MTTR) for software systems that they are working on.

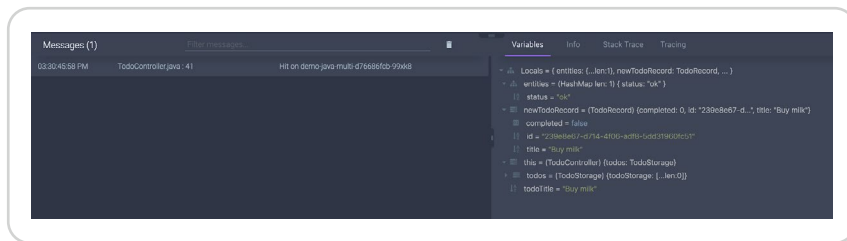
Second, they allow developers to debug an application that's deployed in a staging or production environment.

Third, when debugging seems impossible (and developers can't replicate the bug), it gives developers immediate insight into their application, without stopping it or affecting its performance.

No more frustrated hours spent debugging. Bugs can be solved much more quickly by teams being able to work together.

## Save Time

One of the biggest reasons for adopting a new technology like Rookout is the improvement gains that teams will see around development velocity. Production grade debugging technologies like Rookout can help teams to improve their mean time to repair (MTTR) for software systems that they are working on. That's accomplished through Rookout's technology which allows developers to collect snapshots of debug data instantly from live running applications.



Developers spend huge amounts of time tracking down defects in test and production environments. The cost to fix defects increases exponentially depending on how far along the SDLC the bug is found which means that any time savings when resolving defects later in the SDLC can be of huge value.

Wouldn't it be amazing to be able to debug an application that's deployed in a staging or production environment just like you debug an application in a local debugger within your IDE? That's exactly what Rookout allows you to do. The typical process teams take when they identify a defect in a production environment is to replicate the application in a lower environment such as development or staging in order to properly debug and dive deeper into the issue. With a technology like Rookout, debugging can now happen directly in the environment where the error occurs while getting a familiar debugging experience.



*Production grade debugging technologies help teams to improve their mean time to repair (MTTR) for software systems that they are working on*



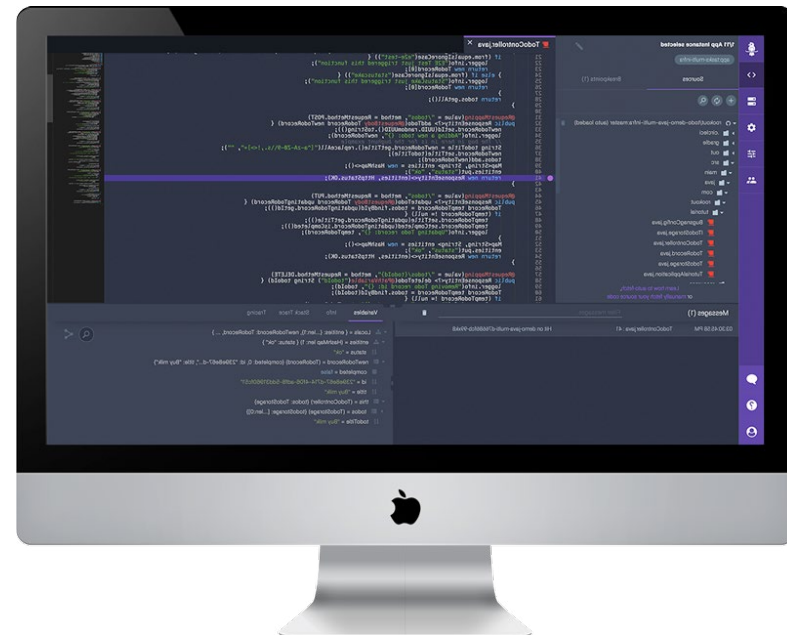
## Debug the Impossible

Have you ever encountered one of those defects that just seems impossible to debug? There are certain bugs that no matter how hard you try to figure out the solution, it seems to elude you and makes you want to bang your head against your keyboard!

We have talked to many organizations that have defects in production environments which they haven't been able to fix for multiple years. Yes, years! Replicating the application in a lower test environment doesn't help as there are often bugs that seem to only be reproducible in the native environment where they occur.



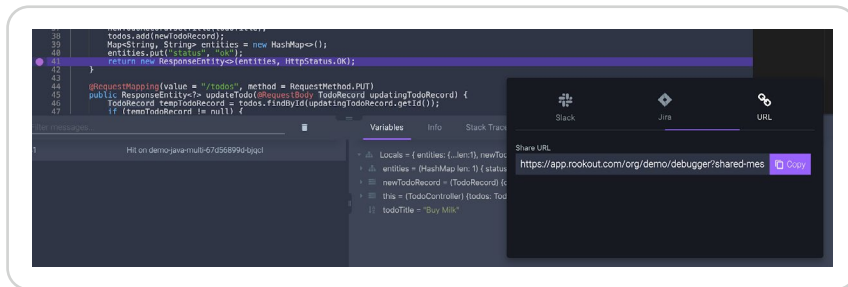
*Production grade debugging solutions give you immediate insight into the values of local variables, stack traces, as well as contextual tracing information all without stopping the application or affecting its performance.*



For these types of situations, a production grade debugging solution like Rookout can add tremendous value. Rookout's non-breaking breakpoints give you immediate insight into the values of local variables, stack traces, as well as contextual tracing information all without stopping the application or affecting its performance. This is especially helpful in cases where you think the issue may be coming from a 3rd party component. If you have the source code from those 3rd party components, Rookout allows you to set non-breaking breakpoints within those code bases just like you would in your own code.

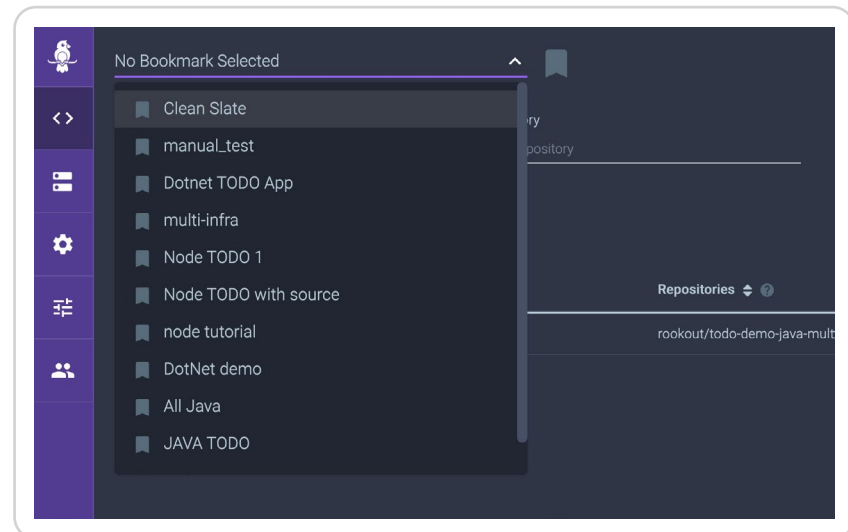
## Improve Developer Communication

Debugging is oftentimes a solo endeavor which can lead to many hours (or days) of frustration before finally solving the problem. Bugs can usually be solved much more quickly by working together with other developers to gain multiple perspectives and insights into the issue. Rookout makes it easy to share captured snapshot data with other developers on your team in order to collaborate more effectively.



As you can see from the screenshot above, Rookout makes it easy to share any debug message captured with a shareable URL. Messages can even be automatically sent to a Slack channel or attached to a JIRA ticket so that developers can better collaborate using existing tools in their environment. In addition, developers can configure Rookout to send captured snapshot messages to any system within their environment which supports a REST API call.

Every developer using Rookout gets their own workspace where they can connect their source code repositories and set non-breaking breakpoints, allowing developers to debug the same application at the same time. Since Rookout's non-breaking breakpoints are real-time and dynamic, developers can even set breakpoints on the same line of code and view the collected data at the same time in their own workspace. Debug sessions including connected source code repositories and sets of application instance filters can be saved with a bookmark so that developers can quickly and easily share debugging configurations for specific applications or environments.



## Wrapping Things Up

Improving developer effectiveness and productivity is at the top of the list of most organizations these days. With many new technology companies fighting for the top spot in their industries, any edge in delivering value is almost always welcomed. Rookout is a must have tool for development teams that want to improve their productivity and save time when debugging application issues in development, staging, or production environments. Bugs that have haunted you for ages now have newfound hope of being solved. Developers can work closely together to solve bugs with a common debugging solution across multiple deployment environments and programming languages.

If you're interested in finding out how Rookout can increase your team's productivity, find out more at <http://www.rookout.com>.

BOOK A DEMO

