

The
Pragmatic
Programmers

Debugging TypeScript Applications

Build Web Apps
That Don't Break



B
E
T
A

Andrey Ozornin
edited by Kelly Talbot

This extract shows the online version of this title, and may contain features (such as hyperlinks and colors) that are not available in the print version.

For more information, or to purchase a paperback or ebook copy, please visit <https://www.pragprog.com>.

Copyright © The Pragmatic Programmers, LLC.

Contents

	Change History	?
	Acknowledgements	?
	Welcome	?
1.	Printing and Interpreting Errors	?
	Setting Up for the Book	?
	Getting Started with Debugging	?
	Fixing Common Errors	?
	Wrapping Up	?
2.	Setting Guardrails with Throw and Catch	?
	Fail-Fast and Fault-Tolerant Systems	?
	Advanced Logging	?
	Wrapping Up	?
3.	Prioritizing Bugs	?
	Estimating Bug Severity	?
	Estimating Bug Reach	?
	Estimating Bug Fix Time	?
	Prioritizing Bugs Using RICE	?
	Prioritizing Beyond Methodologies	?
	Wrapping Up	?

4.	Finding the Root Cause	?
	Finding the Reproduction Steps	?
	Using Systems of Root-Cause Analysis	?
	Using the Scientific Method	?
	Finding the suspect commit using git bisect	?
	Wrapping Up	?
5.	Designing Software That Doesn't Break	?
	Write Clean Code	?
	Let Your Computer Help You	?
	Assure Quality with Automated Testing	?
	Ship with Confidence	?
	Wrapping Up	?
6.	Following the Data Flow	?
	Defining What "Fixed" Means	?
	Overcoming Blank Console Paralysis	?
	Getting to Know the Sources Panel	?
	Navigating the Code	?
	Using Breakpoints	?
	Improving the Fix	?
	Using Source Maps	?
	Wrapping Up	?
	Bonus Exercise	?
7.	Debugging Asynchronous Operations	?
	Network Panel in Chrome Developer Tools	?
	Debug Parallel Operations	?
	Logging on a Webpage	?
	Wrapping Up	?
8.	Make Your Code Debuggable	?
	Setting Up Sentry	?
	Monitoring Errors	?
	Structured Logs	?
	Adding Performance Traces	?
	Wrapping Up	?