



Testable JavaScript: Ensuring Reliable Code

Von Mark Ethan Trostler

audiobook / *ebooks / Download PDF / ePub / DOC



 Download

 Read Online

Produktinformation -Verkaufsrank: #377981 in eBooksVerffentlicht am: 2013-01-16Erscheinungsdatum: 2013-01-16File Name: B00B1WLE92 | File size: 15.Mb

Von Mark Ethan Trostler : Testable JavaScript: Ensuring Reliable Code before purchasing it in order to gage whether or not it would be worth my time, and all praised Testable JavaScript: Ensuring Reliable Code:

KundenrezensionenHilfreichste Kundenrezensionen3 von 3 Kunden fanden die folgende Rezension hilfreich. Guter blick ber Testing und Debugging in JavaScriptVon Inhalt:"Code is for people": Das Buch beginn zunchst mit der Frage, was "testable code" berhaupt bedeutet und was es fr Vorteile mit sich bringt. Danach folgt ein Exkurs ber Code-Komplexitt, wie man sie misst und wie man sie mit z.B. durch lose Kopplung von Komponenten oder event-orientierte Architekturen reduzieren kann. Danach geht es direkt "zur Sache": Wie schreibt man gute Unit-Tests und was ist dabei zu beachten? Wie geht man mit Abhngigkeiten zu anderen Komponenten oder asynchroner Ausfhrung um? Wie misst und verbessert man die Testabdeckung des eigenen Codes (Code Coverage)?Mit Unit-Tests alleine ist es grade bei Web-Applikationen alleine nicht getan, deswegen beschftigt sich das folgende Kapitel mit Integration Tests (z.B. mit Silenium oder CasperJS), Performance- und Load-Testing im Browser und Messen des Ressourcenverbrauches.

Anschließend folgt eine ausführliche Besprechung der unterschiedlichen Debugging-Möglichkeiten, sowohl in den gängigen Browsern, für Node.js-Applikationen oder auch für Mobile Anwendungen auf Android und iOS. Den Abschluss bildet die Automation von Tests mit Continuous Integration-Systemen wie Jenkins. Bewertung: "Testable Javascript" gibt einen umfassenden Überblick über das strukturierte Schreiben, Testen und Debuggen von JavaScript. JavaScript-Applikationen werden immer komplexer, daher wird auch der Aspekt der Codequalität, des Testens und Debuggens immer wichtiger. Es ist schön, dass ein Buch über Testing sich nicht nur auf Unit-Tests beschränkt, sondern den ganzen Testablauf aller (Frontend) Komponenten beleuchtet. Ein wenig gestört hat mich allerdings, dass Trostler (als ehemaliger Yahoo-Mitarbeiter) in vielen Beispielen hauptsächlich auf deren Toolchain eingeht (YUI, YUI Event Hub, YUI Test, YUI Code Coverage), die außerhalb des Unternehmens in der JavaScript-Welt keine große Verbreitung finden. Ein paar Beispiele in marktüblichen Frameworks (und wenn nur zum Vergleich von Implementierungen) wären schön gewesen. Gerade im Testing-Bereich tut sich momentan in JavaScript einiges, so dass sich der Leser auch zusätzlich über aktuelle Entwicklungen informieren sollte. Ergänzend empfiehlt es sich ein paar aktuelle Tools, wie z.B. JSComplexity.com oder etwa Istanbul für Code Coverage (statt einer umständlichen Java-Lösung) genauer in Augenschein zu nehmen. Insgesamt hilft dieses Buch beim notwendigen Verständnis der Materie und erleichtert so den Aufbau eines eigenen Test-Setups. 0 von 0 Kunden fanden die folgende Rezension hilfreich. Zusammen mit "JavaScript The Good Parts" von BzIm49... ist das die Standard-Literatur für JavaScript Developer, die Clean Code produzieren möchten. Gehört auf jeden Developer-Desktop (oder auf den Nachttisch).

Kurzbeschreibung One skill that's essential for any professional JavaScript developer is the ability to write testable code. This book shows you what writing and maintaining testable JavaScript for the client- or server-side actually entails, whether you're creating a new application or rewriting legacy code. From methods to reduce code complexity to unit testing, code coverage, debugging, and automation, you'll learn a holistic approach for writing JavaScript code that you and your colleagues can easily fix and maintain going forward. Testing JavaScript code is complicated. This book helps experienced JavaScript developers simplify the process considerably. Get an overview of Agile, test-driven development, and behavior-driven development. Use patterns from static languages and standards-based JavaScript to reduce code complexity. Learn the advantages of event-based architectures, including modularity, loose coupling, and reusability. Explore tools for writing and running unit tests at the functional and application level. Generate code coverage to measure the scope and effectiveness of your tests. Conduct integration, performance, and load testing, using Selenium or CasperJS. Use tools for in-browser, Node.js, mobile, and production debugging. Understand what, when, and how to automate your development processes.