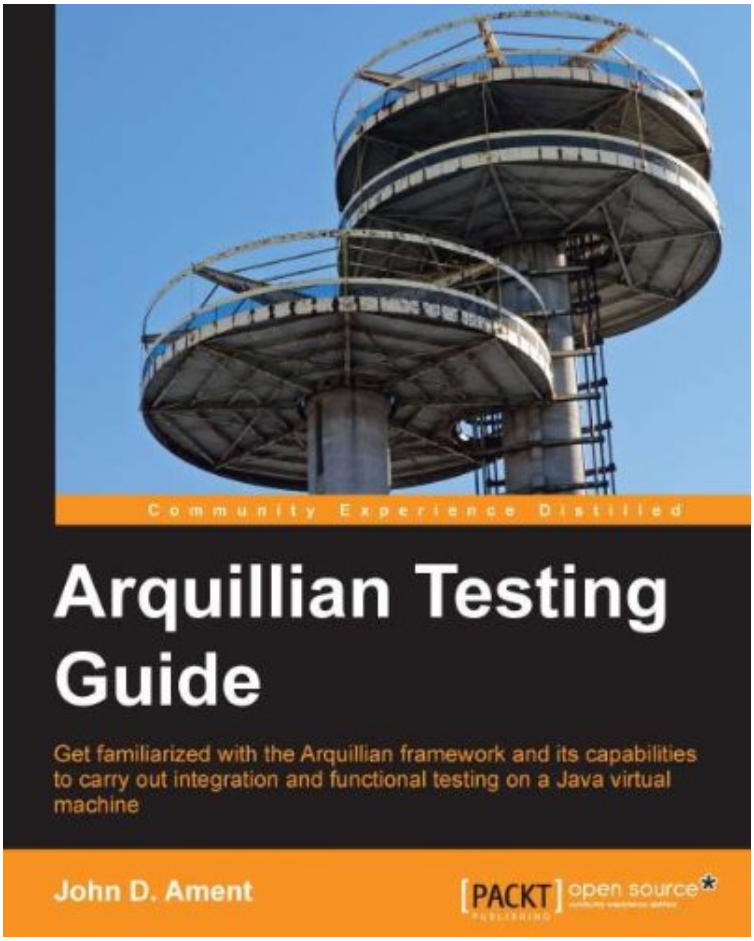
Arquillian Testing Guide

Von John D. Ament
*audiobook | *ebooks | Download PDF | ePub | DOC*



Produktinformation -Verkaufsrang: #343337 in eBooksVerffentlicht am: 2013-04-17Erscheinungsdatum: 2013-04-17File Name: B00BP47WHY | File size: 43.Mb

Von John D. Ament : Arquillian Testing Guide before purchasing it in order to gage whether or not it would be worth my time, and all praised Arquillian Testing Guide:

KundenrezensionenHilfreichste Kundenrezensionen4 von 4 Kunden fanden die folgende Rezension hilfreich. leider konkurrenzlosVon Thomas GrundDas Buch gibt einen recht brauchbaren berblick ber die Thematik. Die Standard-Dokumentation zu Arquillian (insbesondere im Zusammenspiel mit TomEE) ist ja wirklich sprlich. Gut finde ich, da auf die teilweise erheblichen Unterschiede der verschiedenen Ausfhrungsumgebungen (Container) eingegangen wird.Trotzdem scheint mit das Buch keinen umfassenden oder auch nur einfhrenden berblick ber Arquillian zu geben. Selbst so einfache Fragen wie "Wie ldt man eine nicht-leere beans.xml" werden nicht oder nur per Zufall aus Code-Beispielen beantwortet. Hier htte ich mehr System und eine andere Gliederung erwartet. Shrinkwrap htte m.E. nach vorne gehrt, die Details der Container weiter nach hinten.Dazu kommt, da ich die Sprache fr schwer lesbar halte. Manche Stellen sind erkennbar grammatikalisch falsch, also schlecht redigiert. Bei anderen bin ich mir nicht sicher, aber ich stolpere immer wieder. Es gibt englische Bcher, die sich flssiger lesen lassen, was dann Verstndnis und

Vergngen zugute kommt.Ich wrde das Buch wieder kaufen, so lange es kein besseres gibt. Dafr ist Arquillian als Testplattform zu wichtig. Aber ich sehe auch noch viel Potential fr Verbesserungen.

KurzbeschreibungIn DetailIntegration testing sometimes involves writing complex codes. This book introduces you to the capabilities of Arquillian to enable you to write simple code with a broad range of integration tests for java applications. Arquillian Testing Guide serves as an introductory book to writing simple codes for testing java applications. This book will help you to develop richer test cases which can be run automatically while performing rigorous testing of the software. Arquillian Testing Guide introduces you to Arquillians features and capabilities. This book will help you understand the mechanism of creating deployments and test against those deployments. The book begins with basic JUnit test cases beginning with an enterprise test case, which then go on to discuss remote testing. During the course of the book, you will also learn how to mix container and non-container tests into a single test case. By the end of the book, you will have learned how to extend JUnit tests to work with Arquillian and deploy them to a container automatically.ApproachThis book is a tutorial filled with plenty of code examples and strategies to give you many options when building your test structure.Who this book is forThis book is for developers and testers alike. Anyone who has worked with test driven development or building automated test cases will find use in this book. A reader should be familiar with some automation strategies and techniques such as JUnit and should have some exposure to techniques such as mocking.KurzbeschreibungIn DetailIntegration testing sometimes involves writing complex codes. This book introduces you to the capabilities of Arquillian to enable you to write simple code with a broad range of integration tests for java applications. Arquillian Testing Guide serves as an introductory book to writing simple codes for testing java applications. This book will help you to develop richer test cases which can be run automatically while performing rigorous testing of the software. Arquillian Testing Guide introduces you to Arquillians features and capabilities. This book will help you understand the mechanism of creating deployments and test against those deployments. The book begins with basic JUnit test cases beginning with an enterprise test case, which then go on to discuss remote testing. During the course of the book, you will also learn how to mix container and non-container tests into a single test case. By the end of the book, you will have learned how to extend JUnit tests to work with Arquillian and deploy them to a container automatically.ApproachThis book is a tutorial filled with plenty of code examples and strategies to give you many options when building your test structure.Who this book is forThis book is for developers and testers alike. Anyone who has worked with test driven development or building automated test cases will find use in this book. A reader should be familiar with some automation strategies and techniques such as JUnit and should have some exposure to techniques such as mocking.ber den Autor und weitere MitwirkendeJohn D. Ament John D. Ament was born to a beautician and a taxi driver in the warm summer of 1983 in Ridgewood, New York. At the age of six his family moved to the northern suburbs of New Jersey. After graduating from the Rutgers University and working a few short-term IT jobs, he moved to the southern side of New Jersey just outside of Philadelphia, where he has been located for the last seven years. In 2008, he started participating in the open source community. After working with Spring a bit here and there, he started investigating the Seam framework. After finding use of the framework in a few applications, he started participating more and more with the community. Eventually, he became the module lead for a couple of components in Seam 3 and started working more and more with open source technologies. This led to more and more community involvement, including participation in the JMS 2.0 Expert Group. After following through on some test-driven trends, he decided to try out a new framework called Arquillian to help automate some of the testing being performed on a few work-related projects. It surprisingly worked well, to the point of being used to perform all of the automated testing against container deployments of applications he was working on. This drove a lot of passion for the framework and his continued use of the tool today.