



JUnit in Action, Third Edition

By Cătălin Tudose
560 pages

JUnit in Action, Third Edition

Reviewed by Andrew Binstock

March 26, 2021

[Download a PDF of this article](#)

JUnit has become such a popular way of testing code that today it dwarfs all other testing frameworks and tools for Java; its counterparts in other languages also dominate in their respective sectors. In large part, this popularity is attributable to its ease of use and facility at testing small bits of code (unit tests)—and to its role as infrastructure for wider-reaching tests, such as integration tests, system tests, and user-acceptance tests (UATs).

Because of this ease of use, JUnit emerged as the foundation for test-driven development (TDD) and in this capacity, it attained even greater traction in the Java community, eventually burying its principal rival, the well-regarded **TestNG**.

Over the years, JUnit has grown from being a small-scale framework to a much larger endeavor. JUnit 3.x was a compact proposition, 4.x was much larger, and 5.x was so ambitious that the testing platform was completely rearchitected. This book focuses primarily on JUnit 5, with occasional comments and explanations for JUnit 4 counterparts and with a useful explanation of how to transition from JUnit 4 to JUnit 5.

As you can see from the title, I am reviewing the third edition of this book. The current author, Cătălin Tudose, was not an author on either the original 2003 edition or the second edition (published in 2010). In fact, none of the authors of the second edition worked on the first edition. In most cases, this odd transition to new authors would create a significant discontinuity between the editions, but this volume bridges smoothly to the present.

The scope of this book is wide indeed. After reviewing JUnit 4 and introducing JUnit 5 and its architecture, the author goes on a long journey through testing with JUnit. This voyage begins with a discussion of software testing principles and an explanation of the various kinds of testing. Then it proceeds to writing testable code, then testing with stubs and, finally, using mocks. The mocks chapter begins with building your own and using them both to test code and refactor it. The author covers three mocking frameworks: EasyMock, jMock, and Mockito. Those chapters are followed by testing with containers (where the author focuses on Arquillian), then from inside the IDE, with build utilities, and finally as part of continuous delivery pipelines.

The author then examines testing under specific scenarios and frameworks, for example [HtmlUnit](#) and [Selenium](#) for browser interfaces, followed by chapters on testing Spring applications and, separately, Spring Boot apps. He wraps up the section on frameworks with chapters about database applications (JDBC and Hibernate) and REST APIs.

Finally, Tudose closes out the book with two lengthy tutorials on TDD and [behavior-driven development \(BDD\)](#)—the latter using [Cucumber](#) and [JBehave](#). That's a lot to cover and not all of it is dependent on JUnit. Nonetheless, considerable JUnit code is presented and dissected, and the explanations of the code are approachable and generally clear.

While the book text is very readable, I frequently found myself frustrated by the author's tendency to gloss too lightly over JUnit-specific topics. For example, the principal driver of the rewrite of JUnit that became version 5 was architectural. JUnit 5's developers modularized the design and, in the process, they made it considerably easier to extend JUnit's functionality through extension points. This big advance is covered in three pages with most of the text explaining material unrelated to the extension mechanism. In a book entirely dedicated to JUnit, it's reasonable to expect some more depth for this important feature.

In addition, I found omissions of useful features. For example, there is no mention of several annotations that allow you to skip or run tests based on the operating system or on the specific release of the Java runtime: [@EnabledOnOs\(MAC\)](#) and [@EnabledOnJre\(JAVA_8\)](#), respectively. Similarly, the ability to set timeouts for tests (via the [@Timeout](#) annotation) is not mentioned, even though the related assertion [assertTimeout\(\)](#) is explained. The annotation has the distinct benefit that it can be applied to an entire class, whereas the assertion is, of course, assertion-specific. And the author omits the ability to run JUnit tests from the command line via the [ConsoleLauncher](#).

These concerns should not imply that this is a deficient book. Far from it. For readers who simply want to learn how to use JUnit 5 with their site's particular tools and preferred frameworks, *JUnit in Action* fits the bill admirably and it's certainly the best published resource for this use case. For those readers, I can recommend it without reservation.

However, if your aim is to obtain deeper understanding of how JUnit works and to become conversant with the tool's full feature set, then this book is less clearly your first choice. At the moment, the best source for that information remains the [excellent documentation](#) on the JUnit site.



Andrew Binstock

Andrew Binstock ([@platypusguy](#)) was formerly the editor in chief of *Java Magazine*. Previously, he was the editor of *Dr. Dobbs' Journal*. He co-founded the company behind the open-source iText PDF library, which was acquired in 2015. His book on algorithm implementation in C went through 16 printings before joining the long tail. Previously, he was the editor in chief of *UNIX*

Review and, earlier, the founding editor of the *C Gazette*. He lives in Silicon Valley with his wife. When not coding or editing, he studies piano.

Contact

US Sales: +1.800.633.0738
Global Contacts
Support Directory
Subscribe to Emails

About Us

Careers
Communities
Company Information
Social Responsibility Emails

Downloads and Trials

Java for Developers
Java Runtime Download
Software Downloads
Try Oracle Cloud

News and Events

Acquisitions
Blogs
Events
Newsroom

ORACLE | **Integrated Cloud**
Applications & Platform Services



© Oracle | Site Map | Terms of Use & Privacy | Cookie Preferences | Ad Choices