

1. Record Nr.	UNINA9911008920403321
Autore	Leopardi Andrea
Titolo	Testing Elixir : effective and robust testing for Elixir and its ecosystem / / Andrea Leopardi and Jeffrey Matthias
Pubbl/distr/stampa	[Raleigh, North Carolina] : , : The Pragmatic Programmers, LLC, , [2021] ©2021
ISBN	9781680508925 168050892X 9781680508932 1680508938
Edizione	[First edition.]
Descrizione fisica	1 online resource (256 pages)
Disciplina	005.14
Soggetti	Debugging in computer science Elixir (Computer program language)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references.
Nota di contenuto	Cover -- Table of Contents -- Acknowledgments -- Andrea Leopardi -- Jeffrey Matthias -- Introduction -- Why Do We Need a Book for Testing in Elixir? -- Who This Book Is For -- How to Read This Book -- About the Code -- Online Resources -- 1. Unit Tests -- Defining the Unit in Unit Test -- Testing with ExUnit -- Organizing Your Tests -- Creating Comprehensive Test Coverage -- Testing Pure Functions -- Refactoring Toward Pure Functions -- Isolating Code -- Wrapping Up -- 2. Integration and End-to-End Tests -- What Is an Integration Test? -- Testing Against Real External Dependencies -- Dependency Doubles -- Interfacing to External Dependencies with Behaviours -- Test Doubles: Stubs, Mocks, and Fakes -- The Hidden Benefits of Dependency Doubles -- Testing the Actual Interaction with Services -- End-to-End Tests -- Wrapping Up -- 3. Testing OTP -- Testing a GenServer -- Controlling the Life Cycle of OTP Processes in Tests -- Testing Periodic Actions -- Testing Singleton Resources -- Testing Resiliency -- Wrapping Up -- 4. Testing Ecto Schemas -- Testing Your Schema Through Changesets -- Refactoring to Increase Test Maintainability -- Creating a SchemaCase for Shared Test Code --

Testing an Ecto Schema as a Data Validator -- Testing an Ecto Schema for Database Interactions -- Testing Your Schema Through Database Calls -- Setting Up Sandbox Mode -- Wrapping Up -- 5. Testing Ecto Queries -- Creating a Factory to Help with Setup -- Adding a DataCase to Help with Setup -- Testing Create -- Testing Read -- Testing Update -- Testing Delete -- Wrapping Up -- 6. Testing Phoenix -- The Role of Phoenix in Your Application -- Testing JSON-Based APIs -- Testing Server-Rendered HTML Applications -- Testing Phoenix Channels -- Wrapping Up -- 7. Property-Based Testing -- Property-Based Testing in Practice in the Elixir Standard Library -- Example-Based Tests.

Introducing Randomness and Property-Based Testing -- Data Generation -- Writing Properties -- Shrinking -- Strategies for Designing Properties -- Stateful Property-Based Testing -- Wrapping Up -- A1. When To Randomize Test Data -- A2. Test Life Cycle -- The Life Cycle of an ExUnit Suite -- Test Cases -- Executing Tests -- An Example and a Drawing -- A3. Test Coverage -- Built-In Test Coverage -- Coveralls and the Excoveralls Library -- Bibliography -- Index -- - A - - - - B - - - - C - - - - D - - - - E - - - - F - - - - G - - - - H - - - - I - - - - J - - - - L - - - - M - - - - N - - - - O - - - - P - - - - Q - - - - R - - - - S - - - - T - - - - U - - - - V - - - - W - .

Sommario/riassunto

Elixir offers new paradigms, and challenges you to test in unconventional ways. Start with ExUnit: almost everything you need to write tests covering all levels of detail, from unit to integration, but only if you know how to use it to the fullest - we'll show you how. Explore testing Elixir-specific challenges such as OTP-based modules, asynchronous code, Ecto-based applications, and Phoenix applications. Explore new tools like Mox for mocks and StreamData for property-based testing. Armed with this knowledge, you can create test suites that add value to your production cycle and guard you from regressions. Write Elixir tests that you can be proud of. Dive into Elixir's test philosophy and gain mastery over the terminology and concepts that underlie good tests. Create and structure a comprehensive ExUnit test suite, starting from the basics, and build comprehensive test coverage that will provide safety for refactoring and confidence that your code performs as designed. Use tests to make your software more reliable and fault tolerant. Explore the basic tool set provided by ExUnit and Mix to write and organize your test suite. Test code built around different OTP functionality. Isolate your code through dependency injection and by using Mox. Write comprehensive tests for Ecto projects, covering Ecto as a database tool as well as a standalone data validation tool. Test Phoenix channels from end to end, including authentication and joining topics. Write Phoenix controller tests and understand the concepts of integration testing in Elixir. Learn property-based testing with StreamData from the author who wrote the library. Code with high confidence that you are getting the most out of your test suite, with the right tools that make testing your code a pleasure and a valuable part of your development cycle. What You Need: To get the most out of this book, you will need to have installed Elixir 1.8 or later and Erlang/OTP 21 or later. In order to complete the relevant chapters, you will also need Ecto 3.1 or later, EctoSQL 3.1 or later and Phoenix 1.3 or later.
