

1. Record Nr.	UNINA9911008404803321
Autore	Bussey Stephen
Titolo	Real-Time Phoenix // Bussey, Stephen
Pubbl/distr/stampa	Pragmatic Bookshelf, , 2020
ISBN	9781680507744 1680507745
Edizione	[1st edition]
Descrizione fisica	1 online resource (328 pages)
Collana	The Pragmatic programmers
Disciplina	005.276
Soggetti	Phoenix (Web framework) Web site development Elixir (Computer program language) Web applications Application software - Development
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	"Version: P1.0 (March 2020)."
Nota di bibliografia	Includes bibliographical references.
Nota di contenuto	Cover -- Table of Contents -- Acknowledgments -- Introduction -- Who Should Read This Book? -- About This Book -- About the Code -- Online Resources -- 1. Real-Time is Now -- The Case for Real-Time Systems -- The Layers of a Real-Time System -- Types of Scalability -- Achieving Real-Time in Elixir -- Building Real-Time Systems -- Wrapping Up -- Part I-Powering Real-Time Applications with Phoenix -- 2. Connect a Simple WebSocket -- Why WebSockets? -- Connecting our First WebSocket -- WebSocket Protocol -- Long Polling, a Real-Time Alternative -- WebSockets and Phoenix Channels -- Wrapping Up -- 3. First Steps with Phoenix Channels -- What are Phoenix Channels? -- Understanding Channel Structure -- PubSub -- Send and Receive Messages -- Channel Clients -- Wrapping Up -- 4. Restrict Socket and Channel Access -- Why Restrict Access? -- Add Authentication to Sockets -- Add Authorization to Channels -- Use Authentication from JavaScript -- When to Write a New Socket -- Wrapping Up -- 5. Dive Deep into Phoenix Channels -- Design for Unreliable Connections -- Use Channels in a Cluster -- Customize Channel Behavior -- Write Tests -- Wrapping Up -- 6. Avoid Performance Pitfalls -- Measure Everything -- Keep Your Channels Asynchronous -- Build a Scalable

Data Pipeline -- Wrapping Up -- Part II-Building a Real-Time Application -- 7. Build a Real-Time Sneaker Store -- From Product Requirements to a Plan -- Set Up the Project -- Render Real-Time HTML with Channels -- Update a Client with Real-Time Data -- Run Multiple Servers -- Wrapping Up -- 8. Break Your Application with Acceptance Tests -- The Power of Acceptance Testing -- Break Your App Like a User -- Break Your App Like a Server -- Automate Acceptance Tests With Hound -- Wrapping Up -- 9. Build a Real-Time Shopping Cart -- Plan Your Shopping Cart -- Scaffold Your Shopping Cart Channel. Build Your Shopping Cart Channel -- Add Real-Time Out-Of-Stock Alerts -- Acceptance Test the Shopping Cart -- Wrapping Up -- 10. Track Connected Carts with Presence -- Plan Your Admin Dashboard -- On Track with Phoenix Tracker -- Use Tracker in an Application -- Phoenix Tracker Versus Presence -- Scaffold the Admin Dashboard -- Track Shopping Carts in Real-Time -- Assemble the Admin Dashboard -- Load Test the Admin Dashboard -- Wrapping Up -- Part III-Bringing Real-Time Applications to Production -- 11. Deploy Your Application to Production -- The Lay of the Land -- Achieve Scalability with Load Balancing -- Push New Code Safely -- Cluster Your BEAM Nodes Together -- Advanced Phoenix Channel Configuration -- Wrapping Up -- 12. Manage Real-Time Resources -- Getting Comfortable with Elixir's Scheduler -- Manage Your Application's Memory Effectively -- Inspect a Running Application -- Wrapping Up -- Part IV-Exploring Front-End Technologies -- 13. Hands-On with Phoenix LiveView -- Getting Started with LiveView -- Build a LiveView Product Page -- Write Tests for a LiveView -- Wrapping Up -- 14. Single-Page Apps with React -- Manage Channel State in React -- Write Channels as Components -- Hands-On with React -- React Native Channels -- Wrapping Up -- The End of Our Journey -- Bibliography -- Index -- - SYMBOLS - - - DIGITS - - - A - - - B - - - C - - - D - - - E - - - F - - - G - - - H - - - I - - - J - - - K - - - L - - - M - - - N - - - O - - - P - - - Q - - - R - - - S - - - T - - - U - - - V - - - W - - - Y - .

---

## Sommario/riassunto

Give users the real-time experience they expect, by using Elixir and Phoenix Channels to build applications that instantly react to changes and reflect the application's true state. Learn how Elixir and Phoenix make it easy and enjoyable to create real-time applications that scale to a large number of users. Apply system design and development best practices to create applications that are easy to maintain. Gain confidence by learning how to break your applications before your users do. Deploy applications with minimized resource use and maximized performance. Real-time applications come with real challenges - persistent connections, multi-server deployment, and strict performance requirements are just a few. Don't try to solve these challenges by yourself - use a framework that handles them for you. Elixir and Phoenix Channels provide a solid foundation on which to build stable and scalable real-time applications. Build applications that thrive for years to come with the best-practices found in this book. Understand the magic of real-time communication by inspecting the WebSocket protocol in action. Avoid performance pitfalls early in the development lifecycle with a catalog of common problems and their solutions. Leverage GenStage to build a data pipeline that improves scalability. Break your application before your users do and confidently deploy them. Build a real-world project using solid application design and testing practices that help make future changes a breeze. Create distributed apps that can scale to many users with tools like Phoenix Tracker. Deploy and monitor your application with confidence and

reduce outages. Deliver an exceptional real-time experience to your users, with easy maintenance, reduced operational costs, and maximized performance, using Elixir and Phoenix Channels. What You Need: You'll need Elixir 1.9+ and Erlang/OTP 22+ installed on a Mac OS X, Linux, or Windows machine.

---