

1. Record Nr.	UNINA9911009289703321
Autore	Gospodinov Svilen
Titolo	Concurrent data processing in Elixir : fast, resilient applications with OTP, GenStage, Flow, and Broadway / / Svilen Gospodinov
Pubbl/distr/stampa	[Raleigh, North Carolina] : , : The Pragmatic Programmers, LLC, , [2021] ©2021
ISBN	9781680508970 1680508970 9781680508963 1680508962
Edizione	[First edition.]
Descrizione fisica	1 online resource (171 pages)
Disciplina	005.133
Soggetti	Elixir (Computer program language) Application software - Development
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references.
Nota di contenuto	Cover -- Table of Contents -- Foreword -- Acknowledgments -- Introduction -- Who Should Read This Book? -- About This Book -- About the Code -- Online Resources -- 1. Easy Concurrency with the Task Module -- Introducing the Task Module -- Creating Our Playground -- Starting Tasks and Retrieving Results -- Managing Series of Tasks -- Linking Processes -- Meeting the Supervisor -- Understanding Let It Crash -- Wrapping Up -- 2. Long-Running Processes Using GenServer -- Starting with a Basic GenServer -- GenServer Callbacks In Depth -- Building a Job-Processing System -- Introducing DynamicSupervisor -- Implementing a Supervisor -- Naming Processes Using the Registry -- Inspecting Supervisors at Runtime -- Wrapping Up -- 3. Data-Processing Pipelines with GenStage -- Understanding Back-Pressure -- Introducing GenStage -- Building Your Data-Processing Pipeline -- Adding Concurrency with ConsumerSupervisor -- Creating Multi-Stage Data Pipelines -- Choosing the Right Dispatcher -- Wrapping Up -- 4. Processing Collections with Flow -- Analyzing Airports Data -- Creating Flows and Reading Files -- Performing Reduce Concurrently with Partitions --

Using Windows and Triggers -- Adding Flow to a GenStage Pipeline --
Wrapping Up -- 5. Data-Ingestion Pipelines with Broadway --
Processing Ticketing Events -- Broadway Callbacks In Depth --
Batching Messages -- Using GenStage Producers -- Wrapping Up --
Bibliography -- Index -- - SYMBOLS - - - A - - - B - - - C - - - D
- - - E - - - F - - - G - - - H - - - I - - - J - - - K - - - L -
- - - M - - - N - - - O - - - P - - - R - - - S - - - T - - - U -
- - - W -.

Sommario/riassunto

Learn different ways of writing concurrent code in Elixir and increase your application's performance, without sacrificing scalability or fault-tolerance. Most projects benefit from running background tasks and processing data concurrently, but the world of OTP and various libraries can be challenging. Which Supervisor and what strategy to use? What about GenServer? Maybe you need back-pressure, but is GenStage, Flow, or Broadway a better choice? You will learn everything you need to know to answer these questions, start building highly concurrent applications in no time, and write code that's not only fast, but also resilient to errors and easy to scale. Whether you are building a high-frequency stock trading application or a consumer web app, you need to know how to leverage concurrency to build applications that are fast and efficient. Elixir and the OTP offer a range of powerful tools, and this guide will show you how to choose the best tool for each job, and use it effectively to quickly start building highly concurrent applications. Learn about Tasks, supervision trees, and the different types of Supervisors available to you. Understand why processes and process linking are the building blocks of concurrency in Elixir. Get comfortable with the OTP and use the GenServer behaviour to maintain process state for long-running jobs. Easily scale the number of running processes using the Registry. Handle large volumes of data and traffic spikes with GenStage, using back-pressure to your advantage. Create your first multi-stage data processing pipeline using producer, consumer, and producer-consumer stages. Process large collections with Flow, using MapReduce and more in parallel. Thanks to Broadway, you will see how easy it is to integrate with popular message broker systems, or even existing GenStage producers. --
