

1. Record Nr.	UNINA9910380732903321
Autore	Saeed Luqman
Titolo	Introducing Jakarta EE CDI : Contexts and Dependency Injection for Enterprise Java Development / / by Luqman Saeed
Pubbl/distr/stampa	Berkeley, CA : , : Apress : , : Imprint : Apress, , 2020
ISBN	9781484256428 1484256425
Edizione	[1st ed. 2020.]
Descrizione fisica	1 online resource (XIII, 91 p. 68 illus.)
Disciplina	005.133
Soggetti	Java (Computer program language) Computer programming Programming languages (Electronic computers) Java Programming Techniques Programming Languages, Compilers, Interpreters
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di bibliografia	Includes bibliographical references.
Nota di contenuto	Chapter 1: What is Java EE? -- Chapter 2: What is a Java Specification Request (JSR)? -- Chapter 3: What is a Reference Implementation? -- Chapter 4: What is an Application Server? -- Chapter 5: What is Jakarta EE? -- Chapter 6: Why Jakarta EE? -- Chapter 7: Jakarta EE And the Spring Framework -- Chapter 8: The Contexts and Dependency Injection (CDI) API -- Chapter 9: CDI Stereotypes -- Chapter 10: The CDI Ecosystem -- Afterword.
Sommario/riassunto	Discover the Jakarta EE Contexts and Dependency Injection (CDI 2.0) framework which helps you write better code through the use of well-defined enterprise Java-based components and beans (EJBs). If you have ever wanted to write clean Java EE code, this short book is your best guide for doing so: you will pick up valuable tips along the way from your author's years of experience teaching and coding. Introducing Jakarta EE CDI covers CDI 2.0 in detail and equips you with the theoretical underpinnings of Java EE, now Jakarta EE. This book is packed with so much that by the end of it, you will feel confident to use your new-found knowledge to help you write better, readable,

maintainable, and long-lived mission-critical software. You will: Write better code with the Jakarta EE Contexts and Dependency Injection (CDI) framework Work with the powerful, extensible, and well-defined contextual life cycle for components Use CDI's mechanism for decoupling application components through a typesafe event API Build typesafe interceptors for altering the behaviour of components at runtime Harness the well-defined qualifier system for easy isolation of beans Convert almost any valid Java type to a CDI managed bean with CDI's producer mechanism.
