

1. Record Nr.	UNINA9910458020603321
Autore	Alexandre Thomas
Titolo	Scala for Java developers : build reactive, scalable applications and integrate Java code with the power of Scala // Thomas Alexandre ; cover image by Gregoire Alexandre
Pubbl/distr/stampa	Birmingham, England : , : Packt Publishing, , 2014 ©2014
ISBN	1-78328-364-5
Descrizione fisica	1 online resource (282 p.)
Collana	Community Experience Distilled
Disciplina	005.114
Soggetti	Scala (Computer program language) Multiparadigm programming (Computer science) Electronic books.
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	Cover; Copyright; Credits; Foreword; About the Author; Acknowledgments; About the Reviewers; www.PacktPub.com; Preface; Chapter 1: Programming Interactively within Your Project; Advantages of using Scala for Java projects; More concise and expressive; Increased productivity; Natural evolution from Java; Better fit for asynchronous and concurrent code; Learning Scala through the REPL; Declaring val/var variables; Defining classes; Explaining case classes; Operations on collections; Transforming collections containing primitive types; Collections of more complex objects; Filter and partition Dealing with tuplesIntroducing Map; Introducing the Option construct; A glimpse at pattern matching; The map method; Looking at String Interpolation; The groupBy method; The foldLeft method; Summary; Chapter 2: Code Integration; Creating a REST API from an existing database; The sample database; Setting up a Maven project; Creating JPA entities and REST web service; Running and testing the project; Adding a test in Scala; Setting up Scala within a Java Maven project; Scala and Java collaboration; Converting between collection types; JavaBean-style properties; Scala and Java object orientation Scala traits as enhanced Java interfacesDeclaring objects; Introducing companion objects; Handling exceptions; Differences in style between

Java and Scala code; Adjusting the code layout; Naming conventions; Summary; Chapter 3: Understanding the Scala Ecosystem; Inheriting Java Integrated Development Environments (IDEs); Building with Simple Build Tool (SBT); Getting started with SBT; Creating a sample project; Importing the project in Eclipse, IntelliJ IDEA, and NetBeans; Creating a web application that runs on a servlet container; Using sbt-assembly to build a single .jar archive  
Formatting code with Scalariform  
Experimenting with Scala Worksheets; Working with HTTP; Scala's for comprehension; Taking advantage of Typesafe Activator; Creating an application based on activator templates; The REPL as a scripting engine; Summary; Chapter 4: Testing Tools; Writing tests with ScalaTest; BDD-style testing; Functional testing; Mocking with ScalaMock; Testing with ScalaCheck; Summary; Chapter 5: Getting Started with the Play Framework; Getting started with the classic Play distribution; Getting started with the Typesafe Activator; Architecture of a Play application  
Visualizing the framework stack  
Exploring the request-response lifecycle; Handling a request in the controller; Rendering the view; Playing with authentication; Practical tips when using Play; Debugging with Play; Dealing with version control; Summary; Chapter 6: Database Access and the Future of ORM; Integrating an existing ORM - Hibernate and JPA; Making JPA available in Scala; Dealing with persistence in the Play Framework; A simple example using Anorm; Replacing ORM; Learning about Slick; Scaffolding a Play application; Importing test data; Visualizing the database in the H2-browser  
Exploring the code behind the app generation

---

## Sommario/riassunto

This step-by-step guide is full of easy-to-follow code taken from real-world examples explaining the migration and integration of Scala in a Java project. If you are a Java developer or a Java architect, working in Java EE-based solutions and want to start using Scala in your daily programming, this book is ideal for you. This book will get you up and running quickly by adopting a pragmatic approach with real-world code samples. No prior knowledge of Scala is required.

---