

Getting Productive with Scala, Akka, Kafka and Elasticsearch in 5 Days

Developer to Developer Training Series





Course Outline

The 2 day Functional Programming with Java 8 course is aimed at developers who have been using Java but would like to get more out of the language given the new paradigms. They understand Java syntax, language constructs, distinctive objectoriented characteristics but would like to start thinking in a functional way.

This course is designed to help developers think of Java in a functional way. Mutable variables, exceptions, classical input/output, and all other traces of impurity are eliminated. Building up from first principles and extending all the way to incremental input and output, developers would be able to express every concept using only pure functions. This leads to beautiful code and deep insights into the nature of computation.

The course has a practical focus, mixing presentation with indepth handson labs and exercises.



Proposed Structure

Day 1	Day 2
First brush	Case classes and pattern matching
<ul style="list-style-type: none">◆ Introducing the syntax◆ Primitive and simple types◆ Control constructs◆ Traits◆ Objects◆ Functions	<ul style="list-style-type: none">◆ Structural recursion◆ Kinds of pattern◆ Matching with Lists and Options◆ Pattern matching vs. higherorder functions◆ Pattern matching in unexpected places◆ Pattern matching generalized
Introducing Collections	Functional programming indepth
<ul style="list-style-type: none">◆ Mutability & immutability◆ Lists & Buffers◆ Sets & Maps◆ Arrays◆ For comprehensions◆ Interoperability with Java	<ul style="list-style-type: none">◆ Byname parameters◆ Local methods◆ Recursion and tail recursion◆ Partial functions◆ Currying◆ Partially applied functions

	<ul style="list-style-type: none"> ◆ The power of folding
Objectoriented programming in Scala	Day 3
<ul style="list-style-type: none"> ◆ Objects & modules 	<ul style="list-style-type: none"> ◆ Test Driven Development with Actors
<ul style="list-style-type: none"> ◆ Traits and mixin composition 	<ul style="list-style-type: none"> ◆ Actor lifecycle
<ul style="list-style-type: none"> ◆ Self types 	<ul style="list-style-type: none"> ◆ Fault Tolerance with Actors
<ul style="list-style-type: none"> ◆ Parameterized and abstract types 	<ul style="list-style-type: none"> ◆ Messaging with Akka
<ul style="list-style-type: none"> ◆ Structural types 	<ul style="list-style-type: none"> ◆ Understanding Dispatchers
	<ul style="list-style-type: none"> ◆ Understanding Routers
	<ul style="list-style-type: none"> ◆ Distributed communication with Akka
	<ul style="list-style-type: none"> ◆ Clustering basics
Day 4	Day 5
Kafka	Elasticsearch
<ul style="list-style-type: none"> ◆ Kafka Architecture 	<ul style="list-style-type: none"> ◆ Analyse
<ul style="list-style-type: none"> ◆ Partitions 	<ul style="list-style-type: none"> ◆ Visualise
<ul style="list-style-type: none"> ◆ Topics 	<ul style="list-style-type: none"> ◆ Search Data
<ul style="list-style-type: none"> ◆ Replicas 	<ul style="list-style-type: none"> ◆ Queries
<ul style="list-style-type: none"> ◆ Producers & Consumers 	<ul style="list-style-type: none"> ◆ Filters
<ul style="list-style-type: none"> ◆ Brokers 	<ul style="list-style-type: none"> ◆ Best Practices
<ul style="list-style-type: none"> ◆ Reading form Kafka 	<ul style="list-style-type: none"> ◆ Full text search
<ul style="list-style-type: none"> ◆ High level consumers API 	<ul style="list-style-type: none"> ◆ Indexes
<ul style="list-style-type: none"> ◆ Simple/Direct Consumer API 	<ul style="list-style-type: none"> ◆ Mapping features of Elasticsearch
<ul style="list-style-type: none"> ◆ Producer API 	<ul style="list-style-type: none"> ◆ Installation and deployment
<ul style="list-style-type: none"> ◆ Sync Producers 	
<ul style="list-style-type: none"> ◆ Message Acknowledgement 	
<ul style="list-style-type: none"> ◆ Batching Messages 	
<ul style="list-style-type: none"> ◆ Keyed and Non-Keyed Messages 	
<ul style="list-style-type: none"> ◆ Configuration Settings 	
<ul style="list-style-type: none"> ◆ Consumer Configuration Settings 	
<ul style="list-style-type: none"> ◆ Producer Configuration Settings 	
<ul style="list-style-type: none"> ◆ Producer Configuration Settings 	



Course Prerequisites

To benefit from this course you should have 1-2 years experience with an object-oriented (eg. Java, C#) or functional (eg. Haskell, Scheme) programming language and a good general understanding of object-oriented or functional programming language concepts. Knowledge of any distributed system is a plus.

- ◆ For more information on the course or a discussion on your custom need, send a mail to info@knoldus.com