Akka with Scala

This course introduces experienced Scala developers to the reactive Akka toolkit. The combination of hands-on work and exercises in this course provide the perfect environment to best learn to use Akka with Scala.

Participants

- Developers with basic knowledge of java or scala,
- Developers with a familiarity of Reactive Architecture
- Developers who want to develop resilient, event-driven, scalable applications
- Architects who want to have hands-on experience building Reactive Akka applications

Benefits

- Developers gain knowledge and skills to design fault-tolerant apps using Akka
- Production readiness - create asynchronous, event-driven systems

Outline

1 day:

1. Intro
2. Getting Started with Scala Programming
   a. Introduction to Scala
   b. Scala advantages
   c. Working with Scala
   d. Running our first program
   e. SBT Tool
3. First Steps in Scala
4. OO Basics
   a. Ex - Define a Class
   b. Ex - Define Class Parameters
   c. Ex - Promote Class Parameters I
   d. Ex - Promote Class Parameters II
   e. Ex - Define a Field
   f. Ex - Define a Method
   g. Ex - Define a Operator
   h. Ex - Use Default Arguments
   i. Ex - Use Packages
   j. Ex - Check Preconditions
   k. Ex - Define Case Classes
5. Testing
   a. Tests: Group Exercise
6. Collections and Functional Programming Basics
   a. Functions basics
   b. Higher-Order-Programming
   c. Closoures
7. Collections & common methods
   a. Ex - Use a Sequence
   b. Ex - Use map
   c. Ex - Use flatMap
   d. Ex - Use filter

8. For Loops and For Expressions
    a. Ex - Use for-expressions
9. Inheritance and Traits
    a. Ex - Override toString
    b. Ex - Define an ADT
    c. Ex - Use a Trait
10. Pattern Matching
    a. Ex - Use Match Expressions
    b. Use Patterns: Group Exercise
11. Dealing with Optional Values
    a. Ex - Use Option
12. Handling Failure
    a. Ex - Use Try
13. Testing in Scala
    a. The why and what of TDD
    b. ScalaTest
    c. ScalaMock – a native library to mock objects

3 Days

1. Introduction to Akka
    - Why Akka is Reactive
    - Akka's single model for Concurrency, Distribution, Fault Tolerance
2. Actor Basics
    - The Actor Model
    - Anatomy of an Actor
    - Actors and Mutability
    - Actor Systems
    - Creating/Implementing Actors and Behaviors
    - Sending/Forwarding Messages
    - Sender
    - Child Actors
    - Actor Selections
    - Actor State
    - Scheduler
3. Testing Actors
    - Synchronous Unit Testing with TestActorRef
    - Asynchronous Unit Testing with TestProbe

4. Actor Lifecycle
    - Starting/Stopping Actors
    - Lifecycle Hooks
    - Death watch
5. Fault Tolerance

- Let it Crash design philosophy
- Supervision
- Supervision Directives
- Restart Hooks
- Self Healing

6. Routers and Dispatchers
    - Concurrency vs Parallelism
    - Routers and Routing Strategies
    - Group Routers vs Pool Routers
    - Dispatchers and Dispatcher Types
7. Modifying Actor Behavior
    - Become and Unbecome
    - Stash
8. Ask Pattern
    - Ask Pattern and Pipe Pattern
9. Akka Extensions
    - Creating and using Akka Extensions
10. Finite State Machine
    - Using the FSM DSL to implement Finite State Machines
11. Akka streams & Reactive Programming
12. Akka HTTP
13. Akka Persistence
14. Akka Remoting & cluster
15. Akka with websockets