AKKA FRAMEWORK - 2 DAYS

By Dr. Vishwanath Rao

Introduction, environment Setup and a first application

- The Akka framework
  - The Actor model (definition & origins)
  - Fault tolerance
  - Location transparency
  - Scaling
  - Actors In Akka
- Setting up a Scala/Maven environment
- A first application in Akka
  - Create the project
  - Defining message Classes
  - Defining actor Classes
  - Defining the execution Class
- Architecture & configuration files

Actors & Typed Actors

- Anatomy of an Actor
- Creating Actors (default constructors, custom constructors, actor hierarchy)
- Messaging Models
  - Sending messages
    - Fire and forget
    - Send and receive
  - Receiving messages
  - Replying to messages
  - Forwarding messages
- Stop Actors
- Kill Actors

Fault Tolerance - Actor Lifecycle & State

- Let it crash
- Supervision & supervisor strategy
  - One for one
  - All for one
- Lifecycle callbacks
- Receiving Messages

- Online/Offline state
- Hotswap: Become / Unbecome (& stash)
- Finite State Machine FSM
    - States
    - Behaviour

Concurrency

- Blocking vs event driven API
- Using futures & promises

Dispatchers & Routes

- Dispatchers
    - Dispatcher as a pattern
    - Executor & Dispatchers
    - Types of dispatcher / Which to use when
        - Default dispatcher
        - Pinned dispatcher
        - CallingThread dispatcher
        - Balancing dispatcher
- Routers
    - Types of Routers
    - Router usage
    - Router usage via application.conf
    - Router usage for distributed actors
    - Dynamically resizing routers
    - Custom Router

Clustering

- About the Akka Cluster & the CAP theorem
- Defining a cluster
- Cluster Member Status
- Routing messages to the cluster
- Addressing remote actors

Mailboxes

- Types of mailboxes
- Durable mailboxes
- Circuit breakers

Transactions (time permitting)

Testing

- Writing unit test
- Access the actor reference
- Testing actor behaviour
- Testing exception scenarios

JMX and REST interfaces

- RESTful API

- JMX