# Introduction to Apache Camel Integration Framework

Apache Camel is a lightweight, open-source integration framework mainly used for the Heterogeneous protocol integrations.

**Course Objectives**

To impart practical training to the software developers and architects on **Apache** **Camel** enterpriseintegration, administration, deployment and monitoring across multi-protocol integrations.

Post training the participants will be ready to work with Apache Camel framework in enterpriseintegrations with different protocols and applications.

**Prerequisites**

The participants must be proficient in java object oriented programming and aware of JavaEE enterprise applications, Spring framework and exposure to messaging applications, Soap and REST API services is desirable.

**Target audience**

Maximum 20 participants who want to learn to apply the Apache Camel framework in enterprise applications.

**Training Methodology**

The theoretical topics are discussed interactively and technical details are demonstrated with practical examples.

**Each topic is supplemented with practical demonstrations and exercises for the participants**.

The participants work on the hands on case study exercises which strengthen the concepts learned.

**Software Installations**

The Intel quad core compatible CPU with minimum 8GB RAM and 500GB HDD with Windows 10 64 bit and Adobe Acrobat reader, MySql 5.1 workbench client and Database server and JDK1.8 64 bit to be installed and configured. The zip distributions of Eclipse-2020, Apache Active MQ Server ver. 5.14.3, Apache Tomcat 8.0 Web server and Apache Camel ver. 3.7.x dependencies should be able to download with Maven.

Apache Active MQ Server ver. 5.16.1

[**https://mirrors.estointernet.in/apache//activemq/5.16.1/apache-activemq-5.16.1-bin.zip**](https://mirrors.estointernet.in/apache//activemq/5.16.1/apache-activemq-5.16.1-bin.zip)

**Eclipse-jee-2020-03**

**https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/2020-03/R/eclipse-jee-2020-03-R-incubation-win32-x86\_64.zip**

**The participants must have admin rights on their systems.**

**Live internet connection with reasonable speed and download permissions should be provided in the training room to update the patches/plug-ins.**

**Training duration**: 5 **days**

**Customization Points**

1.The type of applications and programming platforms the members have worked :

   Applications: Web applications, Enterprise applications

   Programming platforms: Java,J2EE

2.The deployment environment for the applications/services.

   Server Deployments, CI/CD

3.The objectives from Apache Camel training

   Objective is to use apache camel to implement adapters or API’s to communicate/process fault management information.

4.Specific services/database the members want to integrate with Camel.

   REST,SOAP,SNMP

5.The tools the members want to integrate with Camel.

  Gradle, Maven, groovy scripting, Spock framework for testing

https://ssl.gstatic.com/ui/v1/icons/mail/images/cleardot.gif**Course Plan**

**This is a tentative training plan**

**The final day wise course plan will be submitted based on the inputs from the client team.**

**Specific Topics**

The development team needs following topics to be covered in training

2.       Basic of Camel Frame Work

3.       How to handle security and authentication

4.       Exception Handling

5.       Trouble shooting techniques

6.       Implementation of different camel components such as http , soap

7.       How to implement pull and push mechanism in apache camel based on interval or size of the events

**Course Outline**

**Day1**

**Java applications in Enterprise**

* J2EE application standards overview
* J2EE Web applications overview
* HTTP messaging in web applications
* The http protocol and data format
* The JSON and XML data formats.
* JDBC API overview
* Spring framework dependency management

**Sharing in Enterprise applications**

* Enterprise applications overview
* Challenges for Integration
  + Multi-protocol implementations
  + Platform language
  + Compatibilities
  + Data formats
  + Native platforms
  + Dependencies

**Integration frameworks overview**

* Spring Integration
* Mule
* Fuse
* ESB implementations
* Apache camel

**JMS Messaging applications**

* Application messaging
* The JMS standard
* Synchronous and Asynchronous messaging
* Point to Point and Publish-Subscribe messaging
* JMS implementations
* Topic and Queues
* JMS Factories and connections
* JMS message transactions
* JMS applications with ActiveMQ Server

**Integration Framework: Apache camel**

* Patterns of Enterprise Integration
* Apache Camel Architecture
* Features of Apache Camel
* Apache Camel Components
* Terminologies
  + Channel
  + End Point
  + Router
  + Camel Context
  + Camel Java DSL and XML configuration
  + Message Aggregator and Splitter
* Messaging Model
* Setting up Apache Camel and Eclipse

**Day2**

**Messaging with Apache camel**

* End points
* Message Routing with camel
* Working with Beans
* Java DSL
* Camel Context runner
* Spring context runner
* Message Transformation
* Camel API and docs tour

**Apache Camel Messaging Components**

* File Channel components
* JDBC Channel component
* JMS component
* HTTP and REST components
* Direct, SEDA and VM Components

**Apache Camel Message Processing**

##### Message end points

##### Camel Message Model

##### Message Exchange

* Message Processor
* Message Translation

##### RouteBuilder and DSL

##### Camel Context

##### Camel in Spring Context

##### Bean Integration

##### Java CDI

**Day3**

**JMS Integration**

##### Messaging Module Overview

##### Camel JMS Components

##### Camel ActiveMQ Component

##### Integration with Apache ActiveMQ

* Routing Rule implementations
* JMS with Apache Active MQ
* File Channel and JMS integration
* Message Routing and logging
* JMS to JDBC
* File to JMS Endpoints

**Protocol Integration**

* File system monitoring
* File to JMS
* JMS to JDBC
* Message router
* Http to JMS
* JMS to REST
* REST to SOAP

**The Message Router**

##### Routing Module

##### Message Router Pattern

##### Content-Based Router

##### Camel - Content-Based Router

##### Message filtering

##### Wire tap method

**SOAP Web Services** **Integration**

##### SOAP Module Overview

##### The Http module

##### Soap service integration

##### Soap and JMS messaging

##### Marshalling and un-marshalling XML Data

**Day4**

**REST API integration**

##### JAX-RS Module Overview

##### Camel RESTful Service Support

##### REST DSL

##### REST services integration

##### REST JMS integration

##### Message aggregation

##### Message splitter

**More with Camel**

* The security and authentication at Route level
* Exception Handling
* Implement pull and push mechanism
* Synchronous and asynchronous messaging.

**Message Transformation**

* Data transform with EIP and Java
* Message translator
* XML transformation with XSL
* Data formats supported in camel
* CSV,JSON and other data formats
* Extract JSON data
* Marshal JSON Data

**Service activator**

* Working with beans
* Select bean methods
* Bean parameter
* JNDIRegistry usage
* ApplicationContext registry

**Day5**

**JPA Service integration**

* Configure Route
* Service Mapping and CamelContext
* Set up JDBC channel end points
* Integrate JDBC and JPA services
* Camel properties configuration
* JPA and JMS integration
* Mongo and JMS integration

**JPA Clients with Camel**

* Concurrency management
* Transaction management
* Error Handling
* Working with Entity Beans

**Transaction management**

* Apache Camel transactions across protocols
* Define the transaction boundaries
* Control the transactions
* JMS/JDBC transactions

**Overview on Deployment, Testing and monitoring**

* Apache Camel Project Deployment
* Camel Testing with JUnit
* Camel with Arquillian for Integration test
* Monitoring & Managing Camel
* Camel integration best practices

**\*\*\*\*\*\*\*\*\***