**Mass Transit .NetCore Framework**

**Total Duration : 5 Days (40 Hours)**

**Prerequisite**

**Participants should have strong knowledge on ASP.NET CORE 3.x and Intermediate Microservices.**

**Topics for Discussion**

1.        Setup/Configuring an Azure Service Bus host with a single consumer and a single saga on different endpoints.  
2.        The middleware pipeline for a MassTransit Bus, and how to add custom serialization through middleware.  
3.        Integration/Component testing for the configuration of the bus, as well as the flow of messages through the bus.  
4.        Endpoints, Topologies, and Conductor.  
5.        Courier, faults, exception handling best practices, and transactional operations.

**Day 1**

Getting Started

Live Coding

Upgrading

Using MassTransit

Configuration

Console App

ASP.NET Core

Transports

Broker Topology

Configuration

Guidance

Azure Service Bus

In Memory

Riders

Event Hub

Mediator

**Day 2**

Messages

Message Names

Message Headers

Correlation

Guidance

Consumers

Consumer

Instance

Handler

Producers

Send

Publish

Message Initializers

Send Headers

Custom Serialization

Debugging common problems

Using Message and Consumer Observers.

**Day 3**

Exceptions

Retry

Retry Configuration

Redelivery

Outbox

Faults

Error Pipe

Dead-Letter Pipe

Requests

Request Consumer

Request Client

Container Configuration

Multiple Requests

Multiple Response Types

Sagas

Automatonymous

Introduction

State Machine

Instance

State

Event

Behavior

Configuration

Event

Ignore Event

Composite Event

Missing Instance

Initial Insert

Completed Instance

Activities

Publish

Send

Respond

Schedule

Request

Custom

**Day 4**

Consumer Sagas

Persistence

Azure Table

Cosmos DB

Entity Framework Core

Redis

Azure Service Bus

Containers

Definitions

Microsoft

MultiBus

Autofac

Using Nested Container

Using a Module

Registering State Machine Sagas

Saga persistence

Castle Windsor

Using Window Installer

Simple Injector

StructureMapers

**Day 5**

Testing

Consumer

Saga

State Machine Saga

Advanced

WireMock

Scheduling

Schedule Messages

Amazon SQS

Azure Service Bus

Hangfire Scheduler

Courier

Activities

Building a routing slip

Executing the routing slip

Monitoring routing slip execution

Subscriptions

Middleware

Receive Pipeline

Circuit Breaker

Rate Limiter

Transaction

Custom

Scoped Filters

Conductor

Configuration

Message Data

Usage

Configuration

Time To Live

Inline Threshold

Repositories

File System

Azure Storage

Monitoring

DiagnosticSource

Application Insights

Performance counters

Connect Endpoint

Observers

Received Messages

Consumed Messages

Types Consumed Messages

Sent Messages

Published Messages

Topology

Message Topology

Publish Topology

Send Topology

Consume Topology

Topology Conventions

Exchanges

RoutingKey

Addressing

Broker Topology

Azure Service Bus

Quickstart

Hub Endpoints

Interop

Complex Hubs

Protocol Dictionary

Sample

Sample-SignalR

Console Sample

Considerations

Message Audit

Principles

Installation

Filters

Metadata factory

Batching

Job Consumers