

HELIDON COURSE CONTENTS (5 days)

By Dr. Vishwanath Rao

Prerequisites

- Proficiency in using a Java IDE such as Eclipse, IntelliJ, or Netbeans
- Fundamental knowledge in Java
- Proficiency in object-oriented programming
- Experience with Maven
- Experience with Git

Course Objectives

By the end of the course, you should be able to meet the following objectives:

- Design of a cloud-native microservice architecture
- Build microservices with MicroProfile and Helidon MP
- Build independent REST services
- Persist data by using Java Persistence API and MicroStream
- Secure your microservices
- Test and check your microservices
- Build and deploy your microservices

Course Content

- Cloud-native microservice architecture
- MicroProfile introduction and overview
- Getting started with Helidon MP
- Injecting dependencies into microservices with Contexts and Dependency Injection
- Create a RESTful web service
 - Create a REST service with JAX-RS and JSON-B
 - Use REST client
 - Consume a RESTful web service with JSON-B and JSON-P
 - Document and filter RESTful APIs by using OpenAPI
 - REactive service: Consume RESTful web services asynchronously
- Configuration
 - Static configuration injection using MicroProfile Config
- Fault tolerance
- Observability
 - MicroProfile Metrics by using Jaeger and MicroProfile OpenTracing
 - MicroProfile Health Check

- Security
 - Authentication and authorization
 - Control user and role access by using MicroProfile JSON Web Tokens (JWT)
 - Validate user input by using bean validation
- Persistence
 - Persist data to a database by using Java Persistence API (JPA)
 - Caching HTTP session data by using JCache and MicroStream
 - Persist data by using MicroStream native object graph persistence
- Testing microservices
- Build and deployment of microservices
- Build native executables by using GraalVM
- Getting started with Helidon SE
 - Reactive programming
 - Get started with Helidon SE and Helidon SE configuration
- Exercises with coding

- Injecting dependencies into microservices with Contexts and Dependency Injection
- Create a RESTful web service
- Fault tolerance
- Observability
- Security
- Persistence
- Persist data by using MicroStream native object graph persistence
- Testing microservices
- Build and deployment of microservices

- Reactive programming
- Get started with Helidon SE and Helidon SE configuration
- Cross-origin resource (CORS) sharing support in Helidon SE
- DB Client - work with databases in a non-blocking way
- Persist data by using MicroStream native object graph persistence
- Build gRPC servers and clients
- Health checks
- Metrics
- GraphQL
- Reactive Messaging
- Reactive Streams
- Security - handle authentication, authorization, and context propagation
- OpenAPI
- Tracing - profile and monitor your applications across multiple

services

- WebClient - handle responses to HTTP requests in a reactive way
- WebServer powered by Netty
- Build native executables by using GraalVM
- Various exercises with coding