**Graphql using Nodejs and Spring Boot Netflix DSG Framework**

Total Duration: 3 Days

Lab Requirements

STS/Eclipse/VSC/Intellij

Java 11/17

Mysql server

Docker desktop

Day 1

Module 1:

Setup Java Spring boot 3.x & GraphQL with DSG Framework

1.Graphql Spring boot DataJPA

2. Graphql Spring boot DataJPA with DSG Framework(High priority)

3. Graphql Spring boot Data JPA DSG with Apollo Client in Reactjs

Build tool

Maven (priority)

Gradle

* Hello GraphQL
* Query Nested Data
* Data Resolver From Gradle Plugin
* GraphQL "Overloading"
* ID Data Type
* Extended Java Scalar Library
* Enumeration
* Interface
* Schema Documentation
* Union
* Mutation
* Http Headers
* Subscription

Module 2:

Create Graphql With Database Schema

* Core Object Types
* Relationship & Query
* Mutation
* Subscription
* Coding Preparation
* Underlying Data (Spring JPA)
* Bean Mapping
* GraphQL Components - Skeleton
* Query Resolver - Problem
* Query Resolver - Item Search

Day 2

* Data Resolver - User Login
* Exception Handling - Theory
* Exception Handling - Code
* Mutation Resolver - Problem
* Mutation Resolver - Solution
* Subscription Resolver – Problem(Kafka/Rabbitmq)
* Subscription Resolver – Solution

Module 3

Securing Graphql Operations

* Mutation Resolver - User Administration
* Securing User Administration 1
* Securing User Administration 2

Day 3

Module 4

Java Client Preparation (Intro)

* Java Client 1
* Java Client 2
* Java Client Using Apollo
* Apollo – Query (Reactjs)
* Apollo – Mutation (Reactjs)
* Apollo – Subscription (Reactjs)
* Swagger
* Postman

Data Loader and n+1 problem

Rate Limiting

Depth Limiting

Module 5

Define Graph Model

* Task: Define Graph Model
* Guided Challenge: Define Graph Model
* Full Solution: Define Graph Model