|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lesson | Slide Deck | Fundamentals | Microservices | Event Driven Design | Deployment | Hands-On |
| 1  (Day 1) | Introduction.pptx | * Introductions * Tooling and Environment * Case Study |  |  | * Primer on Docker and Compose CLI tools | * Setting up our environment with docker and docker-compose. * Connecting to MySQL * Restoring our databases |
| 2  (Day 1) | Software-Engineering-Fundamentals-Part-1.pptx | * Basic Java Program with Gradle * OOP Fundamentals and design patterns * SOLID principles * Test Driven Development * Adding dependencies in Gradle |  |  | * TLS considerations | * Creating a new Gradle project * Design and build a Splunk Client Package |
| 3  (Day 2) | Asynchronous-programming-Part-1.pptx | * Threading Fundamentals * Producer/Consumer Pattern * Worker Queues |  |  |  | * An experiment with Threads and ConcurrentQueues * Design and build a logging utility service |
| 4  (Day 2) | Asynchronous-programming-Part-2.pptx | * Batching/Prefetching from Worker Queues * Timers * Other approaches to Thread management * Futures * Observables and Streams |  |  |  | * Update our Logger to take advantage of pre-fetch using Timers * Update our Logger with Reactive Extensions and asynchronous patterns. |
| 5  (Day 3) | Software-Engineering-Fundamentals-Part-2.pptx | * Dependency Injection * Application Configuration * Application Logging |  |  |  | * Using Sprint IoC for DI in our service * Adding configuration and * error logging to our logging-service |
| 6  (Day3) | Introduction-to-containers.pptx |  |  |  | * Container Fundamentals * Containerizing a JVM application * Deployment Options for Containers | * Create Dockerfile for our logging-service * Build the container * Deploy the container * Stop and delete containers and images |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lesson | Slide Deck | Fundamentals | Microservices | Event Driven Design | Deployment | Hands-On |
| 7  (Day 4) | Introduction-to-Microservices.pptx | * Domain Driven Design | * Introduction to Microservices * Service Architecture for our Northwind Case Study |  | * Production-Ready Checklist | * Identifying boundaries within the larger model |
| 8  (Day 4) | Introduction-to-SpingBoot.pptx | * The Spring Framework | * Spring Boot Applications |  |  | * Creating the Customer-Service with Spring Boot |
| 9  (Day 4) | Clean-Code-Architecture.pptx | * Clean Code | * Clean Architecture |  |  | * Implement the Customer Domain |
| 10  (Day 4) | Restful-Services.pptx |  | * ReST Architecure * Guidelines for ReSTful API design * Spring MVC ReST * Testing ReSTful Services |  |  | * Creating Rest endpoints in our Customer-Service * Test our service with Postman * Test our service with Sprint Boot Test |
| 11  (Day 5) | Data-Access.pptx | * SQL Basics * JDBC * Spring Data JDBC Templates | * Concurrency control: Handling stale state |  |  | * SQL Walkthrough Lab * Adding persistence to our Customer-Service |
| 12  (Day 6) | Observability.pptx |  | * Logging * Metrics * Tracing |  |  | * Configure Splunk’s StatsD endpoint * Add logging, metrics, tracing to our Customer-Service |
| 13  (Day 7) | Event-Driven-Design.pptx |  |  | * Introduction to Event Driven Design * Pub/Sub Pattern * Smart Services. Dumb Pipes. * Spring Messaging |  | * Identify the Events for our Customer-Service * Publish Events from our Customer-Service |
| 14  (Day 8) | Securing-Microservices.pptx | * OWASP Top 10 | * Securing Microservices * API Keys * Authorization Tokens |  |  | * Secure our Customer-Service |
| 15  (Day 8) | Api-Documentation.pptx |  | * OpenAPI Spec * Swagger-UI |  | * Checklist review | * Add documentation to our Customer-Service * Deploy our Customer-Service |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lesson | Slide Deck | Fundamentals | Microservices | Event Driven Design | Deployment | Hands-On |
| 16  (Day 9) | Case-Study.pptx | * Calculated fields in business objects * State Management | Case Study: Order Domain |  |  | * Creating our Order-Service project * Creating our Order-Service Domain Objects |
| 17  (Day 9) | Hibernate.pptx | * ORMs * Hibernate Fundamentals * One-To-Many Mapping: Uni-Directional * Composites | * JPA |  |  | * Implement data access with Hibernate and JPA |
| 18  (Day 10) | Asyncronous-Programming-Part-3.pptx |  | * Async Rest Controllers * “Async all the way!” |  |  | * Implement async in our Order-Service |
| 19  (Day 11) | Reacting-to-Events.pptx |  |  | * Reacting to events * Testing in Event Driven Design |  | * Implement event handling when “OrderShipped” is received. |
| 20  (Day 11) |  |  |  |  |  | * Finishing up and deploying our Order-Service |
| 21  (Day 12) | Building-Resilient-Microservices.pptx |  | * Designing for Failure * Timeouts * Retry. Transient Faults * Circuit Breakers * Bulkheads |  |  | * Adding resiliency to our inter-service calls. |
| 22  (Day 13) | Hibernate.pptx | * Bi-Directional Mapping: Many-to-One, One-To-Many | * Case Study: Catalog |  |  | * Create our Catalog-Service |
| 22  (Day 13) | Synchronizing-Data.pptx |  |  | * Synchronizing data between services * Publishing data changes |  | * Publish data changes from our Catalog-Service. |
| 23  (Day 14) | Synchronizing-Data.pptx |  | * Case Study: Inventory | * Responding to published data changes |  | * Create our Inventory-Service and Respond to catalog data changes. |
| Week 4 | Mulesoft | | | | | |
| Week 5 | Final Project | | | | | |
|  |  |  |  |  |  |  |