**Spring Boot and Microservices - Curriculum**

|  |  |
| --- | --- |
| **Topic** | **Details** |
| **Spring Boot Rest security Microservices Docker** | Introduction |
|  | What is a Microservice? |
|  | Sample Microservices Architecture Download and Install Postman HTTP Client Postman Overview |
|  | Resource and Collection URIs |
|  | HTTP Methods GET, POST, DELETE and PUT |
|  | HTTP Headers Accept and Content Type Setting up Development Environment Install Java Platform(JDK) |
|  | Download and Install Spring Tool Suite(STS) Building RESTful Web Services . |
|  | Introduction |
|  | Creating a New Project |
|  | Creating a new Spring project using Spring Boot Initializr Create Users Rest Controller class |
|  | Adding Methods to Handle POST, GET, PUT, DELETE HTTP requests Running Web Service Application |
|  | Reading Path Variables with @PathVariable annotaion Reading Query String Request Parameters |
|  | Making Parameters Optional or Required Returning Java Object as Return Value |
|  | Returning Object as JSON or XML Representation Set Response Status Code |
|  | Reading HTTP POST Request Body. The @RequestBody annotation. Validating HTTP POST Request Body |
|  | Store Users Temporary Handle HTTP PUT Request Handle HTTP Delete Request Handle an Exception |
|  | Return Custom Error Message Object |
|  | Handle a Specific Exception |
| **Spring Boot Rest security Microservices Docker** | Coverage |
|  | Throw and Handle You Own Custom Exception Catch More Than One Exception with One Method |
|  | Dependency Injection Create and Autowire a Service Layer Class Constructor Based Dependency Injection |
|  | Run Web Service as a Standalone Application Eureka Discovery Service |
|  | Introduction to Eureka Discovery Service Startup Eureka Service Discovery |
|  | Users Microservice |
|  | Introduction to Building a Users Microservice |
|  | Users Microservice - Create new Spring Boot Project Enable Spring Discovery Cloud Client |
|  | Create Users Rest Controller |
|  | Access Users Web Service Endpoint via Eureka Discovery Service Exercise - Create Account Management Microservice |
|  | Account Management Microservice |
|  | Introduction to Building an Account Management Microservice Password Reset - Create a new Spring Boot Project |
|  | Access Account Management Microservice via Eureka Discovery Service Zuul API Gateway |
|  | Introduction to Zuul API Gateway Create a ZUUL API Gateway Project Access Microservices via API Gateway |
|  | –Zuul as a Load Balancer Load Balancer - Introduction |
|  | Starting Up More Microservices Trying How Load Balancer Works |
|  | –H In-Memory Database |
|  | H In-memory Database. Introduction. H Database Console Overview Adding Support for the H Database |
| **Spring Boot Rest security Microservices Docker** | –Users Microservice - Implementing User Sign up Introduction |
|  | Adding method to handle HTTP Post Request Implementing the Create User Request Model class Validating HTTP Request Body |
|  | Application Layers |
|  | Implementing Service Layer Class Create a Shared DTO Class Generate Unique Public User Id Adding Support for Spring Data JPA Implementing User Entity Class |
|  | Implementing Spring Data JPA CRUD Repository Save User Details in Database |
|  | Return Http Status Code |
|  | Implementing Create User Response Model Add Spring Security to Users Microservice Add WebSecurity Configuration |
|  | Encrypt User Password |
|  | Allow only IP address of Zuul API Gateway Adding Support to Return JSON or XML |
|  | –Users Microservice - Implementing User Login Introduction |
|  | Implementing LoginRequestModel |
|  | AuthenticationFilter. Implementing attemptAuthentication() Implementing loadUserByUserName() |
|  | AuthenticationFilter. Implementing successfulAuthentication(). Trying How User Login Works |
|  | Customize User Authentication URL |
|  | –Enable Spring Security on API Gateway Introduction to Spring Security on API Gateway Adding Support for Spring Security and JWT Tokens Enable Web Security in Zuul |
|  | Allow Access to Registration and Login Urls Implementing Authorization Filter |
|  | Accessing Protected Microservices with Access Token |
|  | –Spring Cloud Config Server - Git Backend Introduction to Spring Cloud Config Server Create Your Own Config Server |
|  | Create Private GitHub Repository |
|  | Naming Property Files Served by Config Server |
|  | Configure Config Server to Access Private GitHub Repository Adding Properties File to Git Repository |
|  | Configure Users Microservice to be a Client of Config Server |
|  | Make Zuul Gateway a Client of Config Server |
| **Spring Boot Rest security Microservices Docker** | –Spring Cloud Bus |
|  | Introduction to Spring Cloud Bus |
|  | Add Spring Cloud Bus & Actuator Dependencies Enable the /bus-refresh URL Endpoint Download and Run Rabbit MQ |
|  | Rabbit MQ Default Connection Details Trying how Spring Cloud Bus Works Change default Rabbit MQ Password |
|  | –Spring Cloud Config - File System Backend |
|  | Introduction to Spring Cloud Config File System as a Backend Setting up File System Backend |
|  | ing Values Returned by Config Server Trying how Microservices work |
|  | –Spring Cloud Config - Configuration for Multiple Microservices Introduction |
|  | Shared and a Microservice specific properties |
|  | –Spring Boot Actuator |
|  | Introduction to Spring Boot Actuator |
|  | Add Spring Boot Actuator to API Gateway Trying How It Works |
|  | Enable Actuator for Users Microservice |
|  | –Using MySQL Instead of In-Memory Database |
|  | –Encryption and Decryption |
|  | Introduction to Encryption and Decryption of Configuration Properties |
| **Spring Boot Rest security Microservices Docker** | Add Java Cryptography Extension Symmetric Encryption of Properties |
|  | Creating a Keystore for Asymmetric Encryption Asymmetric Encryption of Properties |
|  | –Microservices Communication |
|  | Introduction to Microservices Communication Albums Microservices Source Code |
|  | Clone Source Code of Albums Microservice A walk through an Albums Microservice Implementing Get User Details |
|  | Make Users Microservice call Albums Microservice Trying how it works |
|  | Feign Web Service Client - Introduction Enable Feign in Spring Boot Project Create Feign Client |
|  | Using Feign Client |
|  | Trying How Feign Client Works |
|  | Enable HTTP Requests Logging in Feign Client Handle FeignException |
|  | Handle Response Errors with Feign Error Decoder Hystrix Circuit Breaker & Feign. Introduction. |
|  | Configure Project to use Hystrix Circuit Breaker Trying How Hystrix Circuit Breaker & Feign work Error Handling with Feign Hystrix FallbackFactory |
|  | –Distributed Tracing with Sleuth and Zipkin |
|  | Introduction to Distributed Tracing with Sleuth and Zipkin Add Spring Cloud Sleuth to Users Microservice |
|  | Checking Trace ID and Span ID in a Console |
| **Spring Boot Rest security Microservices Docker** | Startup Zipkin Server View Traces in Zipkin |
|  | –Aggregating Log Files with ELK Stack |
|  | Introduction to Aggregating Log Files with ELK Stack Configure Microservices to Log into a File Download Logstash |
|  | Configure Logstash to Read Log Files Download and Run Elasticsearch Run Search Query |
|  | Download, Install and Run Kibana View Aggregated Logs in Kibana |
|  | –Secure Eureka Dashboard |
|  | Configure Spring Security to Eureka Server Enable Web Security |
|  | Configure Eureka Clients to use Username and Password Configure Eureka Service URL in Config Server |
|  | Move Username and Password to Config Server Encrypting Username and Password |
|  | –Running Microservices in Docker Containers to AWS EC |
| **Spring Boot Rest security Microservices Docker** | Introduction to Running Microservices in Docker Containers Start up a new Linux Server on AWS EC |
|  | Connect to EC Instance |
|  | Docker Commands Used in this Video Course Install Docker on AWS EC |
|  | Docker Hub Introduction |
|  | Run RabbitMQ Docker Container |
|  | Basic Docker Commands Run, Stop, Start, Remove Containers and Images Create Config Server Docker Image |
|  | Publish Config Server Docker Image to Docker Hub Run Config Server on EC from Docker Hub |
|  | Start New EC Instance for Eureka |
|  | Build Docker Image for Eureka Discovery Service Run Eureka in Docker container |
|  | Elastic IP address for EC Instance Create Zuul Api Gateway Docker Image |
|  | Run Zuul Api Gateway in Docker Container Run Elastic Search in Docker container Run Kibana in Docker Container |
|  | Run Kibana and Elastic Search on the same Network Docker Image for Albums Microservice |
|  | Run Albums Microservice Docker Image on EC Logstash Docker Image for Albums Microservice Run Logstash in Docker container |
|  | Run MySQL in Docker Container |
|  | Make MySQL Docker Container Persist Data on EC Build Users Microservice Docker Image |
|  | Run Users Microservice in Docker container Run Logstash for Users Microservice |
| **Spring Boot Rest security Microservices Docker** | –Multiple Environments Dev, Prod. Introduction |
|  | Preparing Configuration for another environment Creating Beans Based on Spring Boot @Profile used Running Docker Container for Different Environments |
|  | Spring Boot Testing for Microservics , usage of Mockito , Junit |
|  |  |