**Assisted Practice: 2.3 Edge Server and Routing**

This section will guide you to:

* Set up Eclipse to work with Spring Boot using the STS plugin
* Create a Spring Boot application that interacts with the Zuul Edge Server

**Development Environment**

* Eclipse IDE for Enterprise Java Developers v2019-03 (4.11.0)
* Apache Tomcat Server v9.0
* JRE: OpenJDK Runtime Environment 11.0.2
* Spring Boot STS 4
* All other software components are configured automatically by Spring Boot

This lab has eight subsections, namely:

* + 1. Installing the STS Plugin in Eclipse
    2. Creating a Spring Boot Starter Project which is web enabled and configured to work with Zuul services
    3. Configuring SpringEdgeApplication class to act as a dynamic router for Zuul
    4. Configuring application.properties for Zuul
    5. Building the project
    6. Publishing and starting the project
    7. Running the project
    8. Pushing the code to your GitHub repositories

**Step 2.3.1:** Installing the Spring Tool Suite Plugin in Eclipse

* Spring Tool Suite is already installed as an Eclipse plugin in your practice lab. (Refer FSD: Lab Guide - Phase 3)

**Step 2.3.2:** Creating a Spring Boot Starter Project which is web enabled and configured to work with Zuul services

* Open Eclipse
* Go to the **File** menu. Choose **New->Other**
* In the **Wizard** list, select **Spring Boot->Spring Starter Project**
* In **Name,** enter SpringEdge, **Type** as Maven, **Packaging** as Jar, **Group** as com.ecommerce, and **Package** as com.ecommerce
* Click on **Next**
* In the list of **Available** dependencies, scroll down to select **Web->Spring Web Starter**
* Then scroll to select **Spring Cloud Routing->Zuul**
* Click on **Next**
* Click on **Finish**
* This will create the project files in the Project Explorer

**Step 2.3.3:** Configuring SpringEdgeApplication class to act as a dynamic router for Zuul

* In the Project Explorer, expand **SpringEdge->src->main>java>com->ecommerce**
* Double click on **SpringEdgeApplication**
* Add the following code:

**package** com.ecommerce;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.cloud.netflix.zuul.EnableZuulProxy;

**@SpringBootApplication**

**@EnableZuulProxy**

**public** **class** SpringEdgeApplication {

**public** **static** void main(**String**[] args) {

SpringApplication.run(SpringEdgeApplication.class, args);

}

}

**Step 2.3.4:** Configuring application.properties for Zuul

* In the Project Explorer, expand **SpringEdge->src->main>resources**
* Double click on **application.properties**
* Add the following data:

spring.application.name = zuulserver

zuul.routes.products.path = /api/demo/\*\*

zuul.routes.products.url = http://localhost:8080/

server.port = 8111

**Step 2.3.5:** Building the project

* From the **Project** menu at the top, click on **Build**
* If any compile errors are shown, fix them as required

**Step 2.3.6:** Publishing and starting the project

* In the Project Explorer, right click on **SpringEdge->Run As->Spring Boot App**
* Check in the Eclipse Console for the message **Started SpringBootStarterApplication**

**Step 2.3.7:** Running the project

* To run the project, open a web browser and type **http://localhost:8111/api/demo/products**

**Step 2.3.8:** Pushing the code to your GitHub repositories

* Open your command prompt and navigate to the folder where you have created your files.

**cd <folder path>**

* Initialize your repository using the following command:

**git init**

* Add all the files to your git repository using the following command:

**git add .**

* Commit the changes using the following command:

**git commit . -m “Changes have been committed.”**

* Push the files to the folder you initially created using the following command:

**git push -u origin master**