ASSIGNMENT9 - HANDS-ON SPRING BOOT APPLICATION INTEGRATION WITH SPLUNK

PREREQUISITES –

1) **INSTALL SPRING TOOL SUITE -**

   https://spring.io/tools

2) **INSTALL SPLUNK - follow below steps**

3) **INSTALL POSTMAN -** https://www.postman.com/downloads/

Step1 - Splunk download link : https://www.splunk.com/en_us/download

Step2 -

Step 3 – Give username and password

Step 4 – Login http://localhost:8000
Give username and password which you have set in step 3

Step 5 – Create the Data Input for splunk
Go to settings -> Data inputs -> HTTP EVENT COLLECTOR -> Global Settings

Step 6 – Set the below parameters in Global Settings:

Step 7 – Create the Token now [Button will be beside Global settings]

Step 8 – Go to Next -> Select SOURCE TYPE – log4j

Step 9 – Create Index and select that index under "Select Allowed Indexes"

Step 10 – Click on review

Step 11 - Go to settings -> Data inputs -> HTTP EVENT COLLECTOR and copy the token

Step 12 – GO TO SETTINGS -> DATA INPUTS -> EVENT COLLECTOR -> CHECK FOR ALL DETAILS

Step 13 – CLONE THE CODE

https://github.com/DEVOPS-WITH-WEB-DEV/Splunk_Grafana_Setup.git

CHECK FOR EXCLUSION OF DEFAULT SPRING BOOT LOGGING AND ADD LOG4J2 AS DEPENDENCY

STEP 14 – ADD LOGGER STATEMENTS

Create the logger object

```
Import org.apache.logging.log4j.Logger;
```

```java
@Service
public class OrderService {

    Logger logger= LogManager.getLogger(OrderService.class);


    private List<Order> orderList = new ArrayList<>();

    public Order addOrder(Order order) {
        logger.info("OrderService:addOrder execution started..")
        logger.info("OrderService:addOrder request payload {} ",
        order.setOrderPlacedDate(new Date());
```

Step 15 – ADD THE SPLUNK REPOSITORY AND SPLUNK LOGGING DEPENDENCY IN POM.XML

```xml
<repositories>
        <repository>
                <id>splunk-artifactory</id>
                <name>Splunk Releases</name>
                <url>https://splunk.artifactoryonline.com/artifactory/ext-releases-local</url>
        </repository>
</repositories>

<dependencies>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-web</artifactId>
                <exclusions>
                        <exclusion>
                                <artifactId>spring-boot-starter-logging</artifactId>
                                <groupId>org.springframework.boot</groupId>
                        </exclusion>
                </exclusions>
        </dependency>

        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-log4j2</artifactId>
        </dependency>
        <dependency>
                <groupId>com.splunk.logging</groupId>
                <artifactId>splunk-library-javalogging</artifactId>
                <version>1.8.0</version>
        </dependency>
```

Step 16 – MAP YOUR SPLUNK VALUES TO LOG4J2-SPRING.XML FILE

Token :**********
Port : 8088
Source :log4j
Index: order_api_index
Source name : http-event-logs

Step 17 – Open Postman

Add POST method - http://localhost:9090/orders
Add BODY
Add JSON

{

"id":101,

"name":"mobile",

"qty":"1",

"price":30000

}

ADD GET http://localhost:9090/orders

ADD GET http://localhost:9090/orders/101

ADD GET http://localhost:9090/orders/102

Step 18 – Search in SPLUNK with this query

index="order_api_dev" OrderService:getOrder

index="order_api_dev" AND (EXCEPTION OR ERROR)

Search Analytics Datasets Reports Alerts Dashboards Search & Reporti

New Search

Save As Create Table View Close

```
index="order_api_dev" AND (EXCEPTION OR ERROR)
```

Last 15 minutes

✓ **1 event** (5/14/23 9:45:46.000 AM to 5/14/23 10:00:46.000 AM) No Event Sampling Job Verbose Mode

Events (1) Patterns Statistics Visualization

Format Timeline Zoom Out Zoom to Selection Deselect 1 minute per colu

List Format 20 Per Page

Hide Fields All Fields

SELECTED FIELDS
- a host 2
- a source 1
- a sourcetype 1

INTERESTING FIELDS
- a event.logger 1
- a event.message 1

i	Time	Event
>	5/14/23 9:58:52.000 AM	{"host":"localhost","index":"order_api_dev","sourcetype":"log4j","source":"http-event-logs","fields":{"messageFormat":"text"},"event":{" erity":"INFO","logger":"com.javatechie.SplunkDemoApplication","time":"1684038527.122","thread":"http-nio-9090-exec-1","message":"OrderCo oller:getOrder fetch order by id 103"}}{"host":"localhost","index":"order_api_dev","sourcetype":"log4j","source":"http-event-logs","fiel s":{"messageFormat":"text"},"event":{"severity":"INFO","logger":"com.javatechie.service.OrderService","time":"1684038527.123","thread":" p-nio-9090-exec-1","message":"OrderService:getOrder execution started.."}}{"host":"localhost","index":"order_api_dev","sourcetype":"log4 j","source":"http-event-logs","fields":{"messageFormat":"text"},"event":{"severity":"ERROR","exception":"Order not found with id : 103", gger":"org.apache.catalina.core.ContainerBase.[Tomcat].[localhost].[/].[dispatcherServlet]","time":"1684038527.127","thread":"http-nio-9 -exec-1","message":"Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed