

BATCH3.0

12+ REAL TIME PROJECTS

EXTRA DOUBTS CLEARING SESSION

EVERY SAT/SUN 3 HRS

MOCK INTERVIEWS/RESUME GUIDANCE/INTREVIEW QUE

## JOIN BATCH3.0 TODAY AND GET BONUS

- REGISTER TODAY TO GET 35%  
HUGE DISCOUNT ON COURSE
- Go to  
<https://www.praveensingampalli.com> [  
Go to courses]
- Select BATCH3.0
- Apply Coupon **SINGAM35** [ EARLY  
BIRD OFFER - Valid for next few hours  
I\*\*\*\* ]
- Dedicated TELEGRAM group with  
chat option



BATCH3.0 START DATE – MAY 27<sup>TH</sup> 2023 [ 2 MONTHS COURSE ]

1) WEEKEND LIVE ONLINE CLASSES – SATURDAY AND  
SUNDAY [ 3 HOURS EACH SESSION ]

2) 12+ REAL TIME HANDS-ON PROJECTS

**ADDONS FOR BATCH3.0**

- ❖ EXTRA DOUBT CLEARING SESSIONS
- ❖ EXTRA PROJECTS WITH DOCUMENTS
- ❖ JOB SUPPORT VIA FEW EXTRA SESSIONS AFTER  
COMPLETION OF COURSE
- ❖ JOB SUPPORT VIA RESUME TO HR DIRECT CONNECT
- ❖ BONUS: RESUME REVIEWS / MOCK INTERVIEWS / LIVE  
FEEDBACK SHARE TO STUDENTS / INTERVIEW QUESTIONS  
TOOL WISE DISCUSSION

LINK TO JOIN BATCH3.0 - <https://bit.ly/ULTIMATEDEVOPS3> [ CHECKOUT THE  
STUDENTS JOB PLACEMENTS AND TESTIMONIALS ☺☺ ]

## PROJECT1 HANDS-ON SPRING BOOT APPLICATION INTEGRATION WITH SPLUNK

### PREREQUISITES –

- 1) INSTALL SPRING TOOL SUITE
- 2) INSTALL SPLUNK
- 3) INSTALL POSTMAN

**BATCH3.0**

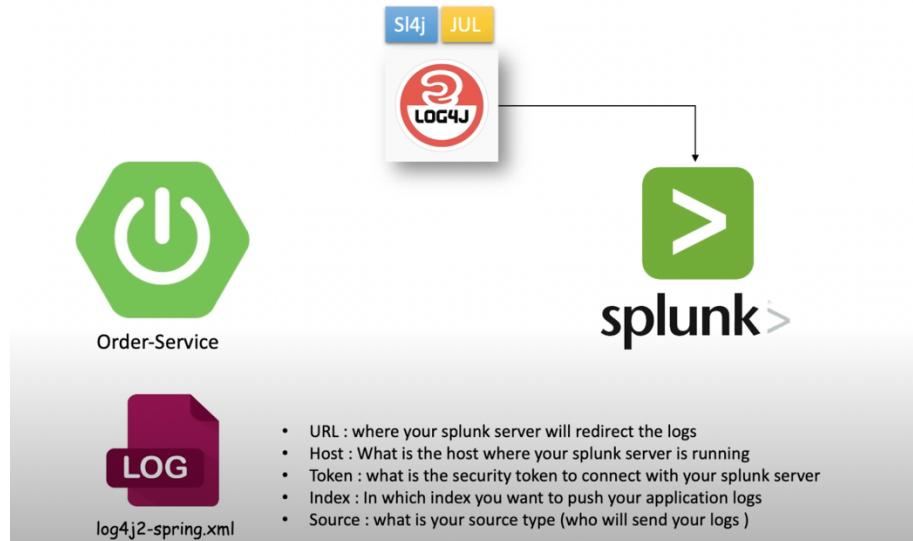
**12+ REAL TIME PROJECTS**

**EVERY SAT/SUN 3 HRS**

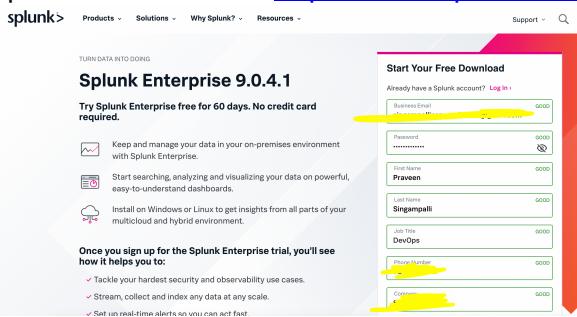
**MOCK INTERVIEWS/RESUME GUIDANCE/INTREVIEW QUE**

**EXTRA DOUBTS CLEARING SESSION**

**MOCK INTERVIEWS/RESUME GUIDANCE/INTREVIEW QUE**



**Step1 - Splunk download link : [https://www.splunk.com/en\\_us/download](https://www.splunk.com/en_us/download)**



**Step2 -**

## Splunk Enterprise 9.0.4.1

Index 500 MB/Day. Sign up and download now. After 60 days you can convert to a perpetual free license or purchase a Splunk Enterprise license to continue using the expanded functionality designed for enterprise-scale deployments.

### Choose Your Installation Package



Windows



Linux



Mac OS

64-bit

Windows 10

Windows Server 2016, 2019, 2022

.msi 453.48 MB

Download Now

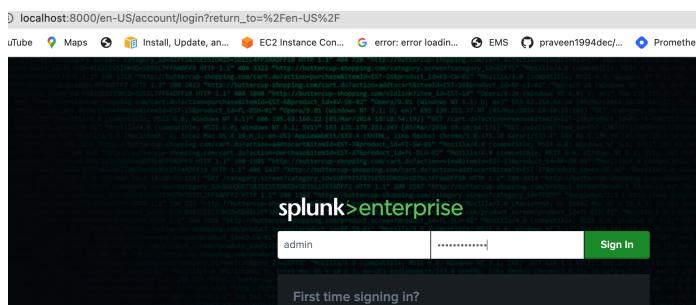


**Step 3 – Give username and password**

```
praveensingampalli — splunk - zsh - 80x24
This appears to be your first time running this version of Splunk.
Splunk software must create an administrator account during startup. Otherwise,
you cannot log in.
Create credentials for the administrator account.
Characters do not appear on the screen when you type in credentials.
Please enter an administrator username: [REDACTED]
```

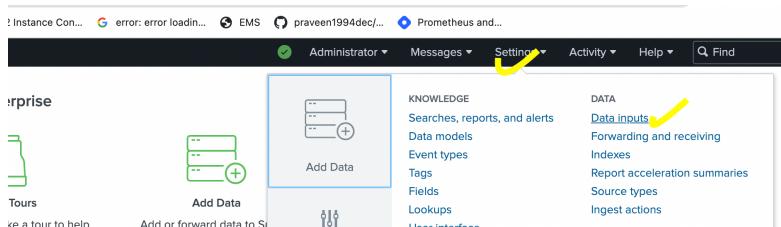
#### Step 4 – Login <http://localhost:8000>

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:

Setting	Value
All Tokens	Enabled
Default Source Type	json
Default Index	order_api_dev
Default Output Group	None
Use Deployment Server	<input type="checkbox"/>
Enable SSL	<input type="checkbox"/>
HTTP Port Number	8088

Cancel Save

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

Configure a new token for receiving data over HTTP. Learn More [Learn More](#)

Name: order-service-logs

Source name override: http-event-logs

Description: optional

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

Add Data     Select Source     Input Settings     Review     Done     < Back     **Review >**

**Input Settings**  
Optionally set additional input parameters for this data input as follows:

Source type: **Select**     **log4j**

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

New Index     **x**

**General Settings**

Index Name: **order\_api\_dev**

Set index name (e.g., INDEX\_NAME). Search using index=INDEX\_NAME.

Index Data Type:  Events     Metrics

The type of data to store (event-based or metrics).

Home Path: optional  
Hot/warm db path. Leave blank for default \${SPLUNK\_DB}/INDEX\_NAME/db.

Cold Path: optional  
Cold db path. Leave blank for default \${SPLUNK\_DB}/INDEX\_NAME/colddb.

Thawed Path: optional  
Thawed/resurrected db path. Leave blank for default \${SPLUNK\_DB}/INDEX\_NAME/thawedb.

Data Integrity Check:  Enable     Disable

Enable this if you want Splunk to compute hashes on every slice of your data for the purpose of data integrity.

**Save**     Cancel

Step 10 – Click on review

Add Data     Select Source     Input Settings     **Review**     Done     < Back     **Submit >**

**Review**

Input Type	Token
Name	order-service-logs
Source name override	http-event-logs
Description	N/A
Enable indexer acknowledgement	No
Output Group	N/A
Allowed indexes	order_api_dev

Default index: order\_api\_dev  
Source Type: log4j  
App Context: launcher

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

Name	Actions	Token Value	Source Type	Index	Status
order-service-logs	Edit   Disable   Delete	9a9cc899-ccdd-479c-a172-9330c35ecf70	log4j	order_api_dev	Enabled

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

Step 13 – CLONE THE CODE

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

```

27
28    <dependencies>
29        <dependency>
30            <groupId>org.springframework.boot</groupId>
31            <artifactId>spring-boot-starter-web</artifactId>
32            <exclusions>
33                <exclusion>
34                    <artifactId>spring-boot-starter-logging</artifactId>
35                    <groupId>org.springframework.boot</groupId>
36                </exclusion>
37            </exclusions>
38        </dependency>
39
40        <dependency>
41            <groupId>org.springframework.boot</groupId>
42            <artifactId>spring-boot-starter-log4j2</artifactId>
43        </dependency>
44        <dependency>
45            <groupId>com.splunk.logging</groupId>
46            <artifactId>splunk-library-javalogging</artifactId>
47

```

STEP 14 – ADD LOGGER STATEMENTS

Create the logger object

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

```
@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.getOrderDetails());
    }
}
```

### Step 15 – ADD THE SPLUNK REPOSITORY AND SPLUNK LOGGING DEPENDENCY IN POM.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>
</dependencies>
```

### 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

BATCH3.0

12+ REAL TIME PROJECTS

EVERY SAT/SUN 3 HRS

EXTRA DOUBTS CLEARING SESSION

MOCK INTERVIEWS/RESUME GUIDANCE/INTREVIEW QUE

{

"id":101,

"name":"mobile",

"qty":"1",

"price":30000

}

http://localhost:9090/orders

The screenshot shows the Postman interface with a POST request to `http://localhost:9090/orders`. The Body tab is selected, displaying the following JSON payload:

```
1 {  
2   "id": 101,  
3   "name": "mobile",  
4   "qty": "1",  
5   "price": 30000  
6 }
```

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

The screenshot shows the Splunk Enterprise search interface with the following search query:

```
index="order_api_dev" OrderService:getOrder
```

The search results show two events from May 14, 2023:

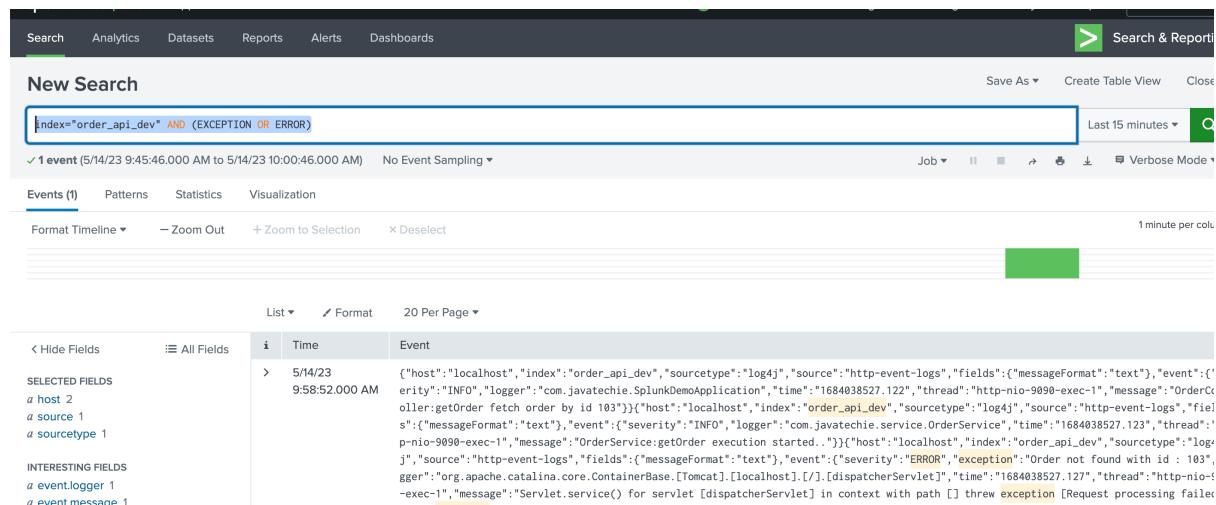
- Event 1: 5/14/23 9:44:08.000 AM to 5/14/23 9:59:08.000 AM
- Event 2: 5/14/23 9:58:52.000 AM

The event details are as follows:

```
5/14/23 9:58:52.000 AM {"host": "localhost", "index": "order_api_dev", "sourcetype": "log4j", "source": "http-event-logs", "fields": {"messageFormat": "text"}, "severity": "INFO", "logger": "com.javatechie.SplunkDemoApplication", "time": "1684038527.122", "thread": "http-nio-9090-exec-1", "message": "OrderService:getOrder fetch order by id 103"} {"host": "localhost", "index": "order_api_dev", "sourcetype": "log4j", "source": "http-event-logs", "messageFormat": "text", "event": {"severity": "INFO", "logger": "com.javatechie.service.OrderService", "time": "1684038527.123", "thread": "http-nio-9090-exec-1", "message": "OrderService:getOrder execution started..."} } {"host": "localhost", "index": "order_api_dev", "sourcetype": "log4j", "source": "http-event-logs", "fields": {"messageFormat": "text"}, "event": {"severity": "ERROR", "exception": "Order not found with id 103"} }
```

index="order\_api\_dev" AND (EXCEPTION OR ERROR)

**BATCH3.0**      **12+ REAL TIME PROJECTS**      **EXTRA DOUBTS CLEARING SESSION**  
**EVERY SAT/SUN 3 HRS**      **MOCK INTERVIEWS/RESUME GUIDANCE/INTREVIEW QUE**



The screenshot shows a Splunk search interface with the following details:

- Search Bar:** Index="order\_api\_dev" AND (EXCEPTION OR ERROR)
- Results Summary:** 1 event (5/14/23 9:45:46.000 AM to 5/14/23 10:00:46.000 AM) No Event Sampling
- Event View:** One event listed in a table.
- Event Fields:**
  - SELECTED FIELDS:** host 2, source 1, sourcetype 1
  - INTERESTING FIELDS:** event.logger 1, event.message 1
- Event Data (Partial):**

```
{"host": "localhost", "index": "order_api_dev", "sourcetype": "log4j", "source": "http-event-logs", "fields": {"messageFormat": "text"}, "event": {"severity": "INFO", "logger": "com.javatechie.SplunkDemoApplication", "time": "1684038527.122", "thread": "http-nio-9090-exec-1", "message": "OrderC\nller:getOrder fetch order by id 103"}}, {"host": "localhost", "index": "order_api_dev", "sourcetype": "log4j", "source": "http-event-logs", "fields": {"messageFormat": "text"}, "event": {"severity": "INFO", "logger": "com.javatechie.service.OrderService", "time": "1684038527.123", "thread": "http-nio-9090-exec-1", "message": "OrderService:getOrder execution started."}}, {"host": "localhost", "index": "order_api_dev", "sourcetype": "log4j", "source": "http-event-logs", "fields": {"messageFormat": "text"}, "event": {"severity": "ERROR", "exception": "Order not found with id : 103", "logger": "org.apache.catalina.core.ContainerBase.[Tomcat].[localhost].[/].[dispatcherServlet]", "time": "1684038527.127", "thread": "http-nio-9090-exec-1", "message": "Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed because [Order not found with id : 103]] with root cause [java.util.NoSuchElementException: Key null is not present]"}}
```