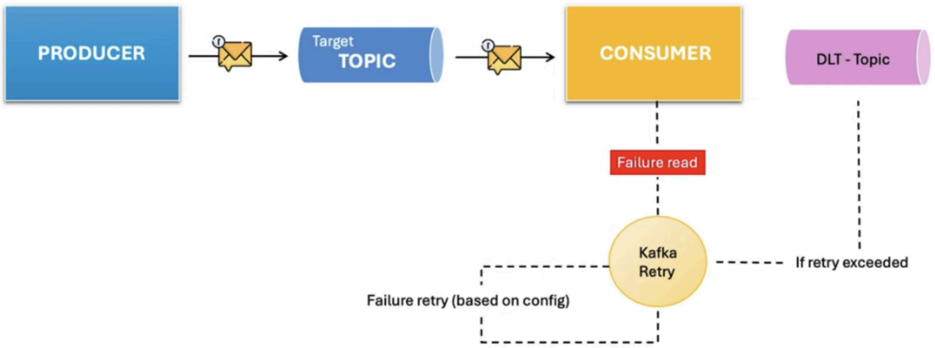
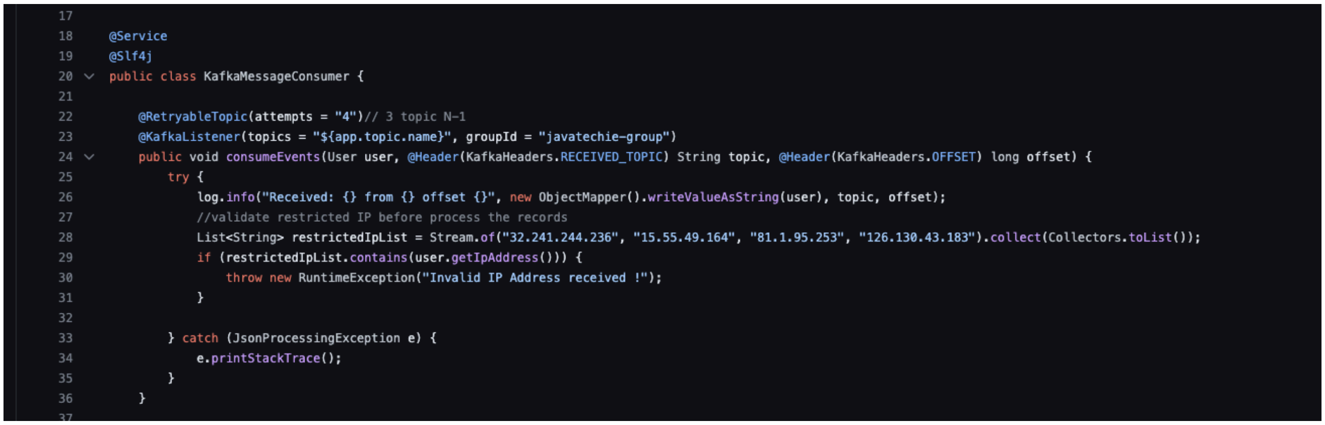
Kafka Error Handling with Spring Boot | Retry Strategies & Dead Letter Topics

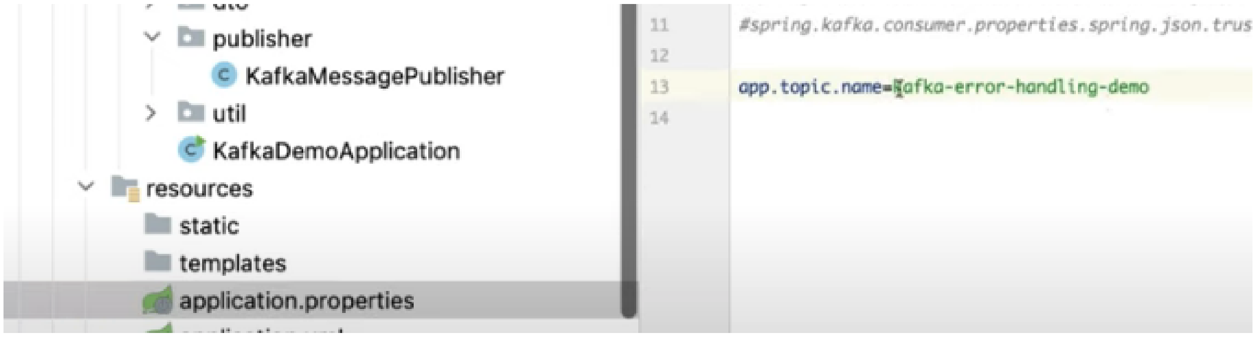


https://github.com/Java-Techie-jt/kafka-error-handling

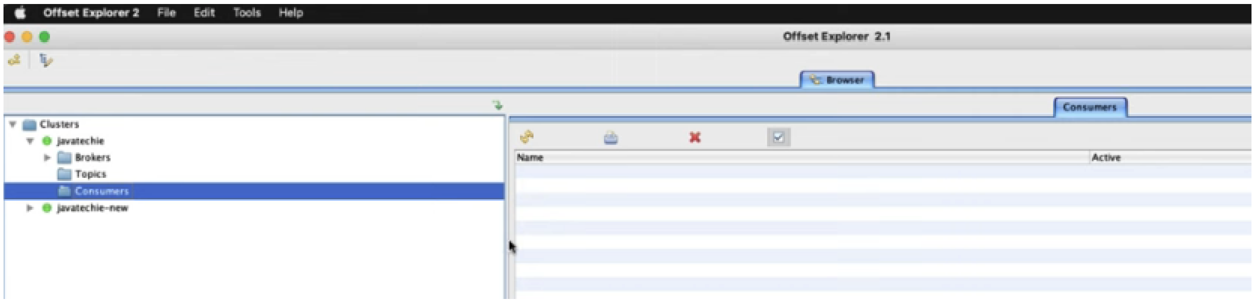


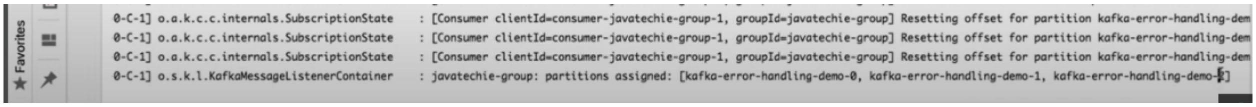
```java

@Service
@Slf4j
public class KafkaMessageConsumer {

    @RetryableTopic(attempts = "4")// 3 topic N-1
    @KafkaListener(topics = "${app.topic.name}", groupId = "javatechie-group")
    public void consumeEvents(User user, @Header(KafkaHeaders.RECEIVED_TOPIC) String topic, @Header(KafkaHeaders.OFFSET) long offset) {
        try {
            log.info("Received: {} from {} offset {}", new ObjectMapper().writeValueAsString(user), topic, offset);
            //validate restricted IP before process the records
            List<String> restrictedIpList = Stream.of("32.241.244.236", "15.55.49.164", "81.1.95.253", "126.130.43.183").collect(Collectors.toList());
            if (restrictedIpList.contains(user.getIpAddress())) {
                throw new RuntimeException("Invalid IP Address received !");
            }

        } catch (JsonProcessingException e) {
            e.printStackTrace();
        }
    }
```

Imitating error scenario in the consumer
Start zookeeper and kafka server and open offset explorer
Topic name

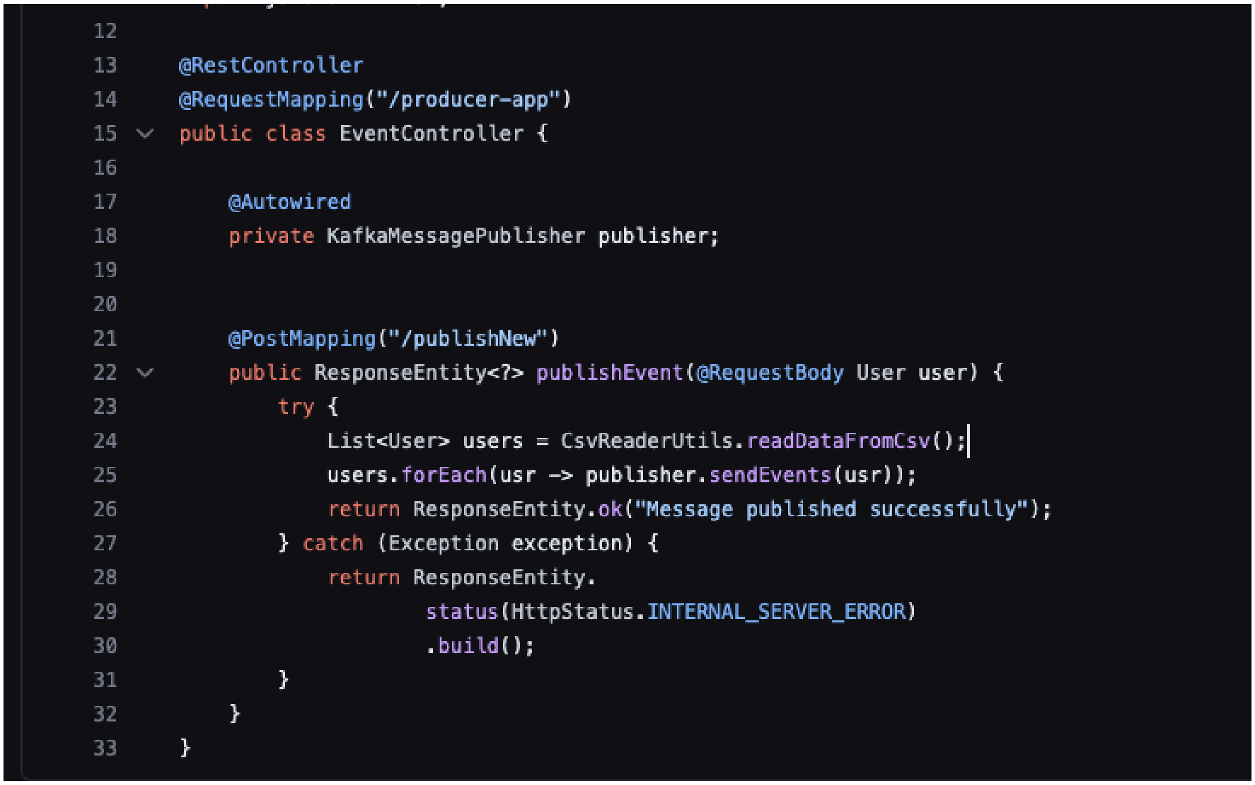


```properties
#spring.kafka.consumer.properties.spring.json.trus

app.topic.name=kafka-error-handling-demo

```

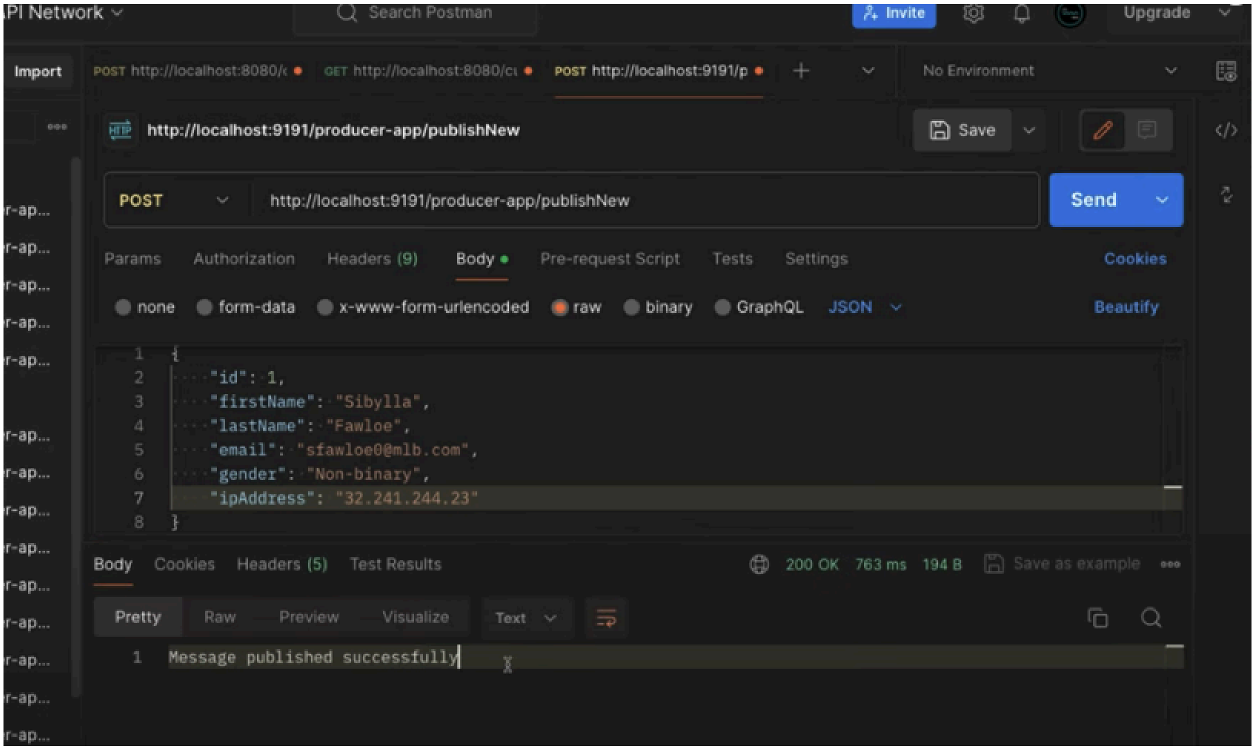




```
0-C-1] o.a.k.c.c.internals.SubscriptionState    : [Consumer clientId=consumer-javatechie-group-1, groupId=javatechie-group] Resetting offset for partition kafka-error-handling-dem
0-C-1] o.a.k.c.c.internals.SubscriptionState    : [Consumer clientId=consumer-javatechie-group-1, groupId=javatechie-group] Resetting offset for partition kafka-error-handling-dem
0-C-1] o.a.k.c.c.internals.SubscriptionState    : [Consumer clientId=consumer-javatechie-group-1, groupId=javatechie-group] Resetting offset for partition kafka-error-handling-dem
0-C-1] o.s.k.l.KafkaMessageListenerContainer    : javatechie-group: partitions assigned: [kafka-error-handling-demo-0, kafka-error-handling-demo-1, kafka-error-handling-demo-2]
```



```

@RestController
@RequestMapping("/producer-app")
public class EventController {

    @Autowired
    private KafkaMessagePublisher publisher;


    @PostMapping("/publishNew")
    public ResponseEntity<?> publishEvent(@RequestBody User user) {
        try {
            List<User> users = CsvReaderUtils.readDataFromCsv();
            users.forEach(usr -> publisher.sendEvents(usr));
            return ResponseEntity.ok("Message published successfully");
        } catch (Exception exception) {
            return ResponseEntity.
                    status(HttpStatus.INTERNAL_SERVER_ERROR)
                    .build();
        }
    }
}
```



http://localhost:9191/producer-app/publishNew

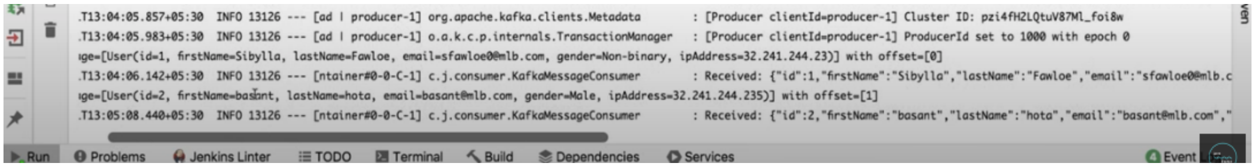
POST http://localhost:9191/producer-app/publishNew

Send

```
{
    "id": 1,
    "firstName": "Sibylla",
    "lastName": "Fawloe",
    "email": "sfawloe0@mlb.com",
    "gender": "Non-binary",
    "ipAddress": "32.241.244.23"
}
```

```
Message published successfully
```

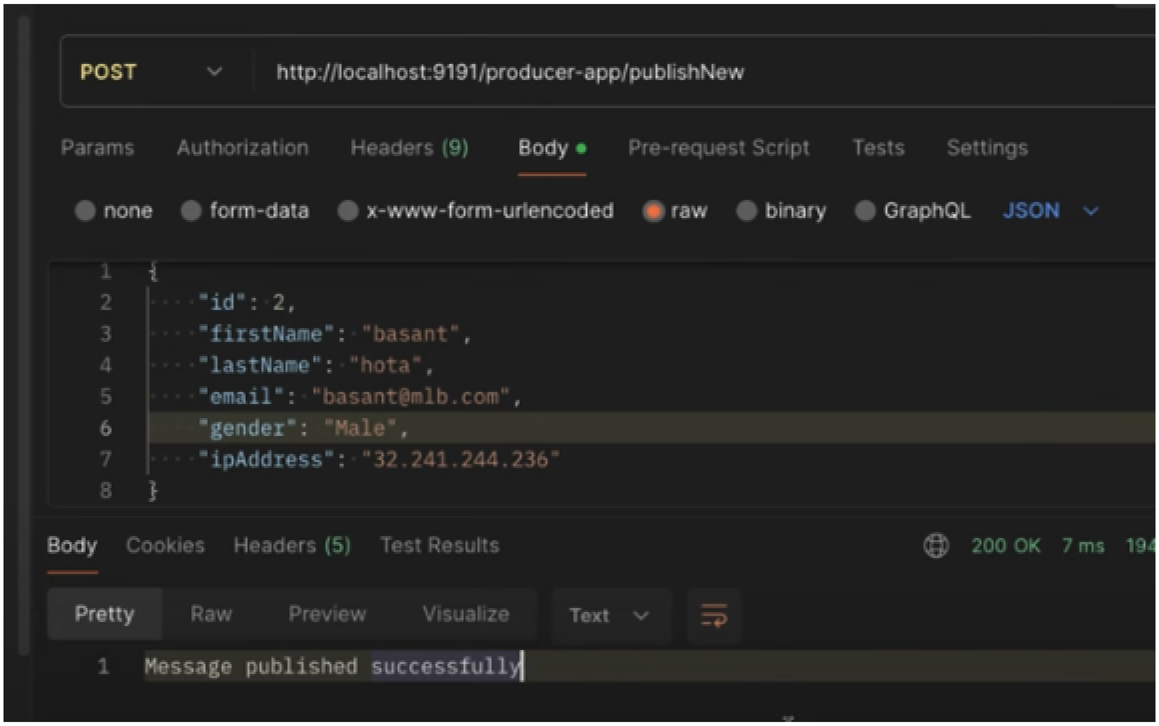
We have success response



```
.T13:04:05.857+05:30  INFO 13126 --- [ad | producer-1] org.apache.kafka.clients.Metadata        : [Producer clientId=producer-1] Cluster ID: pzi4fH2LQtuV87Ml_foi8w
.T13:04:05.983+05:30  INFO 13126 --- [ad | producer-1] o.a.k.c.p.internals.TransactionManager   : [Producer clientId=producer-1] ProducerId set to 1000 with epoch 0
ige=[User(id=1, firstName=Sibylla, lastName=Fawloe, email=sfawloe0@mlb.com, gender=Non-binary, ipAddress=32.241.244.23)] with offset=[0]
.T13:04:06.142+05:30  INFO 13126 --- [ntainer#0-0-C-1] c.j.consumer.KafkaMessageConsumer        : Received: {"id":1,"firstName":"Sibylla","lastName":"Fawloe","email":"sfawloe0@mlb.c
ige=[User(id=2, firstName=basant, lastName=hota, email=basant@mlb.com, gender=Male, ipAddress=32.241.244.235)] with offset=[1]
.T13:05:08.440+05:30  INFO 13126 --- [ntainer#0-0-C-1] c.j.consumer.KafkaMessageConsumer        : Received: {"id":2,"firstName":"basant","lastName":"hota","email":"basant@mlb.com","
```

Two messages submitted successfully. So far we have seen happy scenario.

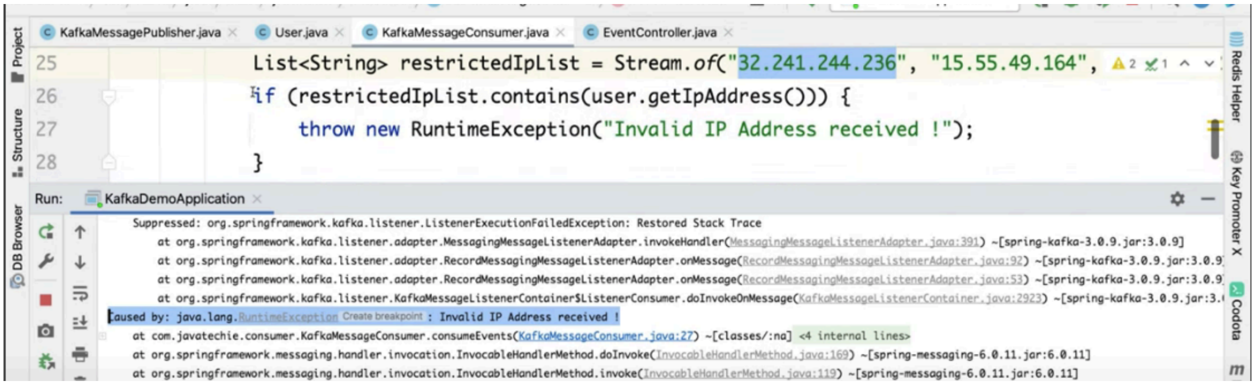
Now try with invalid ip address



http://localhost:9191/producer-app/publishNew

```json
{
    "id": 2,
    "firstName": "basant",
    "lastName": "hota",
    "email": "basant@mlb.com",
    "gender": "Male",
    "ipAddress": "32.241.244.236"
}
```

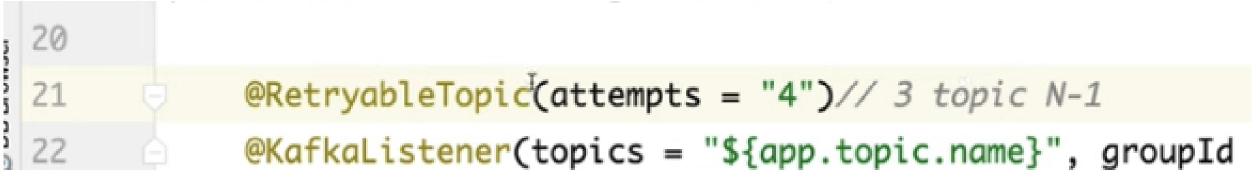
```
Message published successfully
```



```java
                List<String> restrictedIpList = Stream.of("32.241.244.236", "15.55.49.164",
                if (restrictedIpList.contains(user.getIpAddress())) {
                    throw new RuntimeException("Invalid IP Address received !");
                }
```

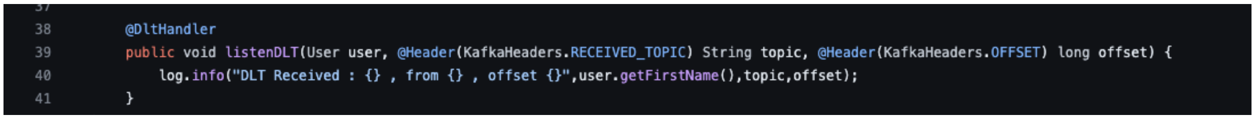
```
    Suppressed: org.springframework.kafka.listener.ListenerExecutionFailedException: Restored Stack Trace
        at org.springframework.kafka.listener.adapter.MessagingMessageListenerAdapter.invokeHandler(MessagingMessageListenerAdapter.java:391) ~[spring-kafka-3.0.9.jar:3.0.9]
        at org.springframework.kafka.listener.adapter.RecordMessagingMessageListenerAdapter.onMessage(RecordMessagingMessageListenerAdapter.java:92) ~[spring-kafka-3.0.9.jar:3.0.9
        at org.springframework.kafka.listener.adapter.RecordMessagingMessageListenerAdapter.onMessage(RecordMessagingMessageListenerAdapter.java:53) ~[spring-kafka-3.0.9.jar:3.0.9
        at org.springframework.kafka.listener.KafkaMessageListenerContainer$ListenerConsumer.doInvokeOnMessage(KafkaMessageListenerContainer.java:2923) ~[spring-kafka-3.0.9.jar:3.
Caused by: java.lang.RuntimeException Create breakpoint : Invalid IP Address received !
    at com.javatechie.consumer.KafkaMessageConsumer.consumeEvents(KafkaMessageConsumer.java:27) ~[classes/:na] <4 internal lines>
    at org.springframework.messaging.handler.invocation.InvocableHandlerMethod.doInvoke(InvocableHandlerMethod.java:169) ~[spring-messaging-6.0.11.jar:6.0.11]
    at org.springframework.messaging.handler.invocation.InvocableHandlerMethod.invoke(InvocableHandlerMethod.java:119) ~[spring-messaging-6.0.11.jar:6.0.11]
```

In realtime it failed due to DB error / mongo error. But we have to take care of that message.



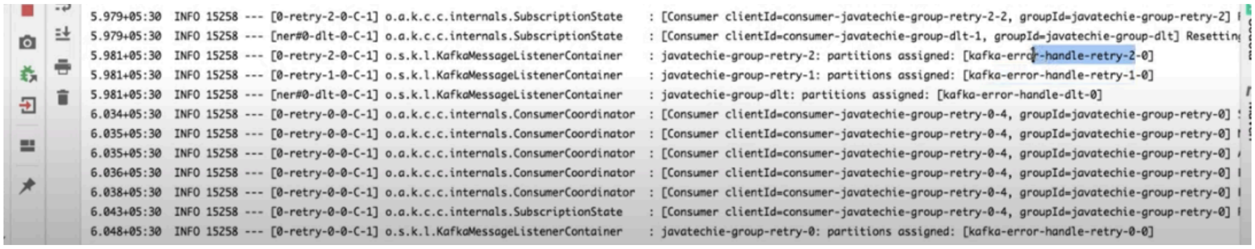
```java
    @RetryableTopic(attempts = "4")// 3 topic N-1
    @KafkaListener(topics = "${app.topic.name}", groupId
```

Let's add @RetryableTopic and @DltHandler method as shown below.

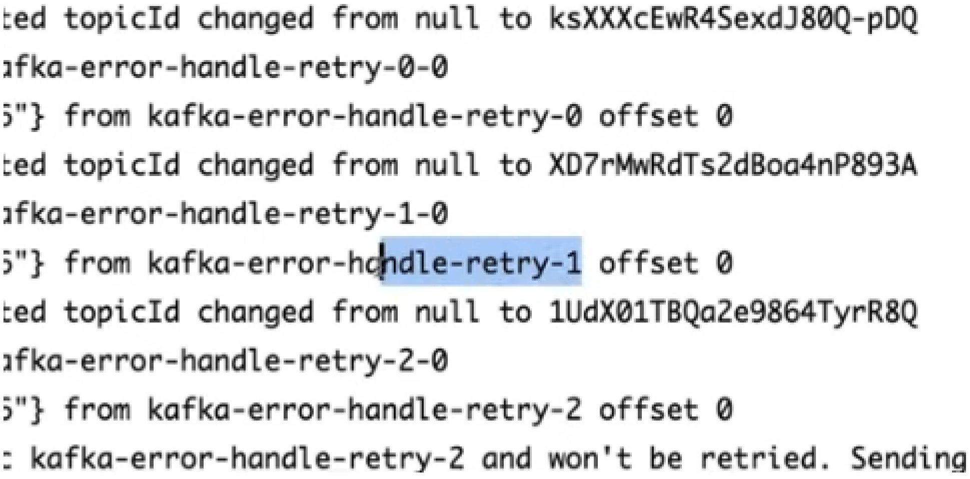


```java
    @DltHandler
    public void listenDLT(User user, @Header(KafkaHeaders.RECEIVED_TOPIC) String topic, @Header(KafkaHeaders.OFFSET) long offset) {
        log.info("DLT Received : {} , from {} , offset {}",user.getFirstName(),topic,offset);
    }
```

We see three retries now after regular attempt.

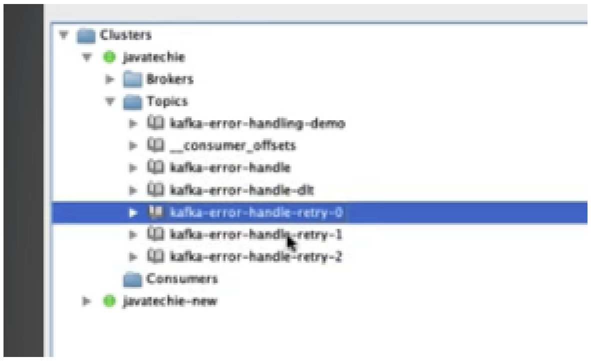


```
5.979+05:30  INFO 15258 --- [0-retry-2-0-C-1] o.a.k.c.c.internals.SubscriptionState     : [Consumer clientId=consumer-javatechie-group-retry-2-2, groupId=javatechie-group-retry-2]
5.979+05:30  INFO 15258 --- [ner#0-dlt-0-C-1] o.a.k.c.c.internals.SubscriptionState     : [Consumer clientId=consumer-javatechie-group-dlt-1, groupId=javatechie-group-dlt] Resettin
5.981+05:30  INFO 15258 --- [0-retry-2-0-C-1] o.s.k.l.KafkaMessageListenerContainer     : javatechie-group-retry-2: partitions assigned: [kafka-error-handle-retry-2-0]
5.981+05:30  INFO 15258 --- [0-retry-1-0-C-1] o.s.k.l.KafkaMessageListenerContainer     : javatechie-group-retry-1: partitions assigned: [kafka-error-handle-retry-1-0]
5.981+05:30  INFO 15258 --- [ner#0-dlt-0-C-1] o.s.k.l.KafkaMessageListenerContainer     : javatechie-group-dlt: partitions assigned: [kafka-error-handle-dlt-0]
6.034+05:30  INFO 15258 --- [0-retry-0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-javatechie-group-retry-0-4, groupId=javatechie-group-retry-0]
6.035+05:30  INFO 15258 --- [0-retry-0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-javatechie-group-retry-0-4, groupId=javatechie-group-retry-0]
6.035+05:30  INFO 15258 --- [0-retry-0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-javatechie-group-retry-0-4, groupId=javatechie-group-retry-0]
6.036+05:30  INFO 15258 --- [0-retry-0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-javatechie-group-retry-0-4, groupId=javatechie-group-retry-0]
6.038+05:30  INFO 15258 --- [0-retry-0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-javatechie-group-retry-0-4, groupId=javatechie-group-retry-0]
6.043+05:30  INFO 15258 --- [0-retry-0-0-C-1] o.a.k.c.c.internals.SubscriptionState     : [Consumer clientId=consumer-javatechie-group-retry-0-4, groupId=javatechie-group-retry-0]
6.048+05:30  INFO 15258 --- [0-retry-0-0-C-1] o.s.k.l.KafkaMessageListenerContainer     : javatechie-group-retry-0: partitions assigned: [kafka-error-handle-retry-0-0]
```



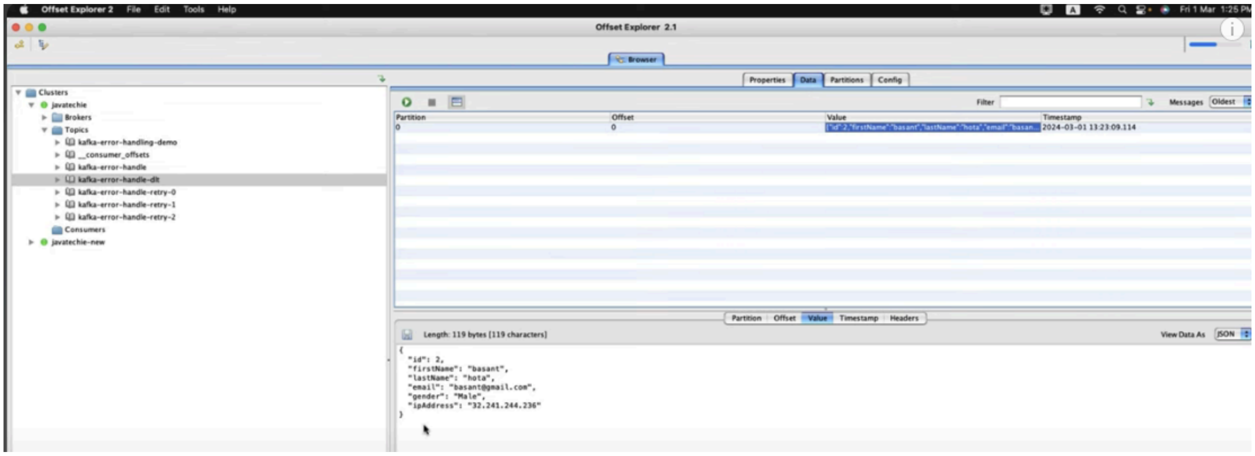
```
ted topicId changed from null to ksXXXcEwR4SexdJ80Q-pDQ
afka-error-handle-retry-0-0
5"} from kafka-error-handle-retry-0 offset 0
ted topicId changed from null to XD7rMwRdTs2dBoa4nP893A
afka-error-handle-retry-1-0
5"} from kafka-error-handle-retry-1 offset 0
ted topicId changed from null to 1UdX01TBQa2e9864TyrR8Q
afka-error-handle-retry-2-0
5"} from kafka-error-handle-retry-2 offset 0
: kafka-error-handle-retry-2 and won't be retried. Sending
```

It creates three retry topics and one dead letter topic at the backend.



```
▼ Clusters
  ▼ javatechie
    ▶ Brokers
    ▼ Topics
      ▶ kafka-error-handling-demo
      ▶ __consumer_offsets
      ▶ kafka-error-handle
      ▶ kafka-error-handle-dlt
      ▶ kafka-error-handle-retry-0
      ▶ kafka-error-handle-retry-1
      ▶ kafka-error-handle-retry-2
      Consumers
  ▶ javatechie-new
```

After three attempts it is not successful then it will reach to Dead letter queue

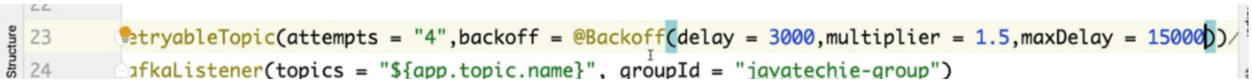


Partition	Offset	Value	Timestamp
0	0	{"id":2,"firstName":"basant","lastName":"hota","email":"basan...	2024-03-01 13:23:09.114

Length: 119 bytes [119 characters]

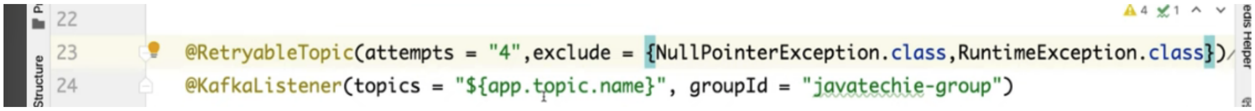
```
{
  "id": 2,
  "firstName": "basant",
  "lastName": "hota",
  "email": "basant@gmail.com",
  "gender": "Male",
  "ipAddress": "32.241.244.236"
}
```

If you want to give time between try play with these values in @RetryTopic annotation.



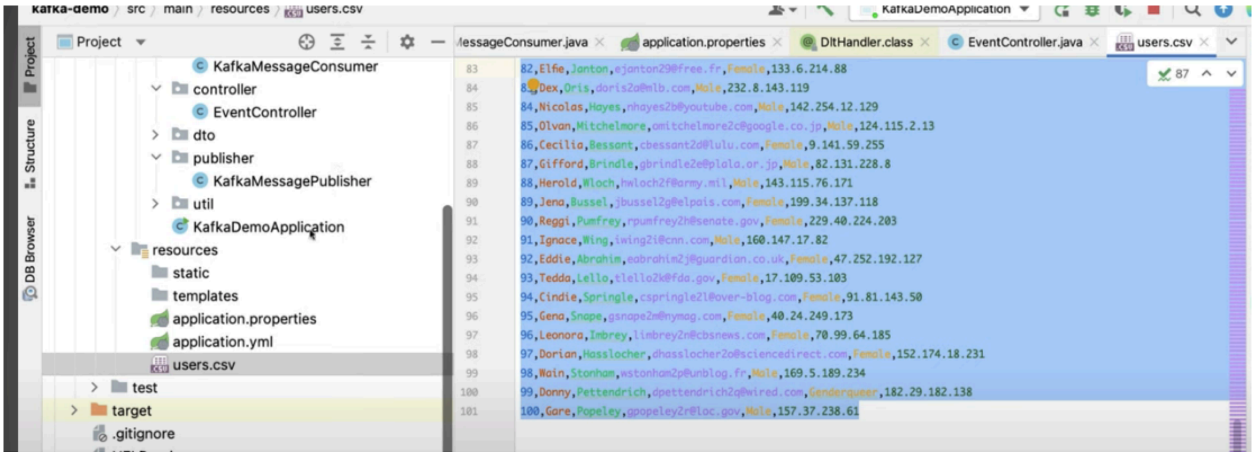
```
etryableTopic(attempts = "4",backoff = @Backoff(delay = 3000,multiplier = 1.5,maxDelay = 15000))
afkaListener(topics = "${app.topic.name}", groupId = "javatechie-group")
```

We can also control for what exception we do not want to retry



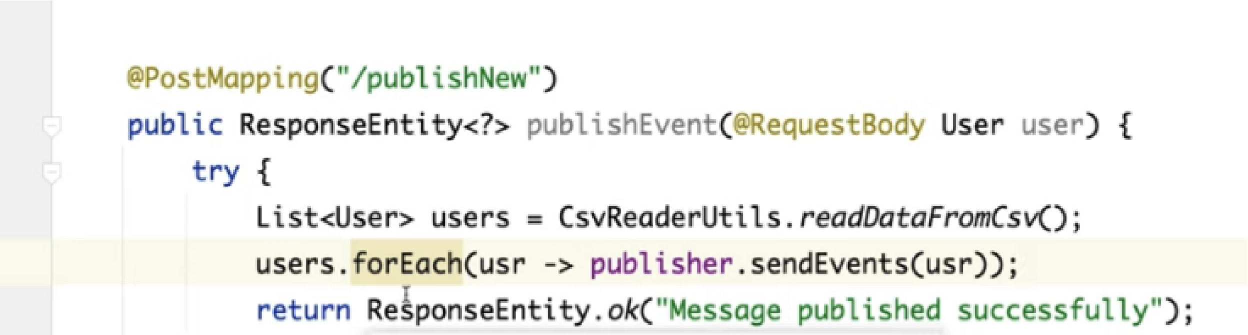
```
@RetryableTopic(attempts = "4",exclude = {NullPointerException.class,RuntimeException.class})
@KafkaListener(topics = "${app.topic.name}", groupId = "javatechie-group")
```

Let us try with bulk without “exclude” and “backoff” attributes with 4 errors records



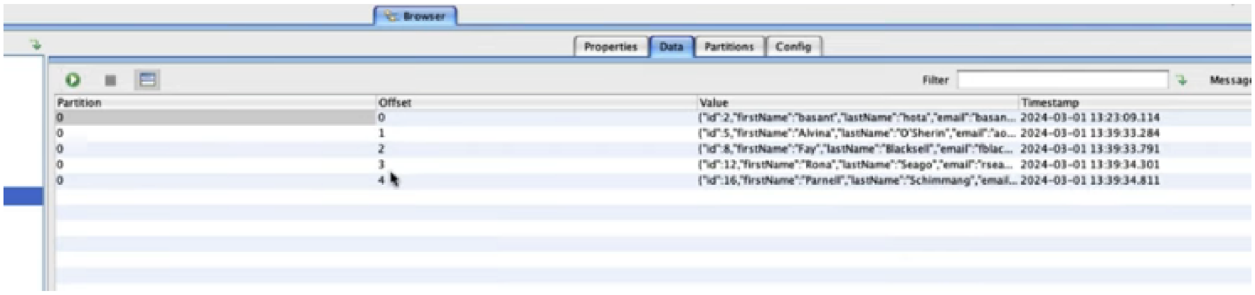
```
82,Elfie,Janton,ejanton29@free.fr,Female,133.6.214.88
83,Dex,Oris,doris2a@mlb.com,Male,232.8.143.119
84,Nicolas,Hayes,nhayes2b@youtube.com,Male,142.254.12.129
85,Olvan,Mitchelmore,omitchelmore2c@google.co.jp,Male,124.115.2.13
86,Cecilia,Bessant,cbessant2d@lulu.com,Female,9.141.59.255
87,Gifford,Brindle,gbrindle2e@plala.or.jp,Male,82.131.228.8
88,Herold,Wloch,hwloch2f@army.mil,Male,143.115.76.171
89,Jena,Bussel,jbussel2g@elpais.com,Female,199.34.137.118
90,Reggi,Pumfrey,rpumfrey2h@senate.gov,Female,229.40.224.203
91,Ignace,Wing,iwing2i@cnn.com,Male,160.147.17.82
92,Eddie,Abrahim,eabrahim2j@guardian.co.uk,Female,47.252.192.127
93,Tedda,Lello,tlello2k@fda.gov,Female,17.109.53.103
94,Cindie,Springle,cspringle2l@over-blog.com,Female,91.81.143.50
95,Gena,Snape,gsnape2m@nymag.com,Female,40.24.249.173
96,Leonora,Imbrey,limbrey2n@cbsnews.com,Female,70.99.64.185
97,Dorian,Hasslocher,dhasslocher2o@sciencedirect.com,Female,152.174.18.231
98,Wain,Stonham,wstonham2p@unblog.fr,Male,169.5.189.234
99,Donny,Pettendrich,dpettendrich2q@wired.com,Genderqueer,182.29.182.138
100,Gare,Popeley,gpopeley2r@loc.gov,Male,157.37.238.61
```

Change the controller method accordingly as shown below.



```
@PostMapping("/publishNew")
public ResponseEntity<?> publishEvent(@RequestBody User user) {
    try {
        List<User> users = CsvReaderUtils.readDataFromCsv();
        users.forEach(usr -> publisher.sendEvents(usr));
        return ResponseEntity.ok("Message published successfully");
```

We received 4 errors



Partition	Offset	Value	Timestamp
0	0	{"id":2,"firstName":"basant","lastName":"hota","email":"basan...	2024-03-01 13:23:09.114
0	1	{"id":5,"firstName":"Alvina","lastName":"O'Sherin","email":"ao...	2024-03-01 13:39:33.284
0	2	{"id":8,"firstName":"Fay","lastName":"Blacksell","email":"fblac...	2024-03-01 13:39:33.791
0	3	{"id":12,"firstName":"Rona","lastName":"Seago","email":"rsea...	2024-03-01 13:39:34.301
0	4	{"id":16,"firstName":"Parnell","lastName":"Schimmang","email...	2024-03-01 13:39:34.811