

Commit f16a194



Klaas-Ritense and **claude** authored 3 weeks ago · ✓ 226 / 227 · Verified



GH-653: Fix sensitive data logging, etc (#497)

* GH-653: Fix sensitive data logging, silent exceptions, and null safety in inbox & SSE mappers

- Downgrade inbox message logging from INFO to DEBUG without payload
- Add WARN-level logging to ValtimoCloudEventManager catch blocks
- Replace non-null assertion with null-safe check in DocumentSseEventManager
- Return null instead of empty string for missing caseDefinitionVersionTag

Co-Authored-By: Claude Opus 4.6 (1M context) <noreply@anthropic.com>

* GH-653: Fix copy-paste bug, null safety, and malformed error messages in RabbitMessagePublisher

- Fix debug log that printed exchange instead of routingKey
- Replace non-null assertion with proper null handling for confirmation result
- Use idiomatic ?.let for returned message check
- Remove stray escaped quotes from exception messages

Co-Authored-By: Claude Opus 4.6 (1M context) <noreply@anthropic.com>

Co-authored-by: Claude Opus 4.6 (1M context) <noreply@anthropic.com>

🔗 [next-minor](#) (#497) · 📦 13.24.0 ... 13.22.0

1 parent [be40474](#) commit f16a194 📄

📁 8 files changed +44 -19 lines changed

↑ Top ⚙️

🔍 Filter files...

- ✓ 📁 backend
 - ✓ 📁 case/src
 - ✓ 📁 main/kotlin/com/ritense


```

38 40      }
39 41      else -> null

```



...e/document/mapper/DocumentSseEventManager.kt



```

@@ -27,6 +27,7 @@ import com.ritense.document.event.DocumentUpdatedSseEvent

```

```

27 27      import com.ritense.inbox.ValtimoEvent
28 28      import com.ritense.valtimo.web.sse.domain.SseEventManager
29 29      import com.ritense.valtimo.web.sse.event.BaseSseEvent

```

```

30 + import io.github.oshai.kotlinlogging.KotlinLogging

```

```

30 31
31 32      class DocumentSseEventManager : SseEventManager {
32 33

```

```

@@ -35,13 +36,20 @@ class DocumentSseEventManager : SseEventManager {

```

```

35 36          DocumentCreated.TYPE -> CaseCreatedEvent()
36 37          DocumentAssigned.TYPE -> CaseAssignedEvent()
37 38          DocumentUnassigned.TYPE -> CaseUnassignedEvent()

```

```

38 -          DocumentUpdated.TYPE -> event.let {
39 -              DocumentUpdatedSseEvent(
40 -                  it.resultId!!
41 -              )

```

```

39 +          DocumentUpdated.TYPE -> {
40 +              val resultId = event.resultId
41 +              if (resultId == null) {
42 +                  logger.warn { "Received DocumentUpdated event without
resultId, skipping" }
43 +                  return null
44 +              }
45 +              DocumentUpdatedSseEvent(resultId)

```

```

42 46      }
43 47
44 48      else -> null
45 49      }
46 50      }

```

```

51 +
52 +      companion object {
53 +          private val logger = KotlinLogging.logger {}
54 +      }

```

```

47 55      }

```



```

  ...per/ConfigurationIssueSseEventManagerTest.kt
  @@ -54,7 +54,7 @@ class ConfigurationIssueSseEventManagerTest {
54 54      }
55 55
56 56      @Test
57 - fun `map should fallback to resultId when result is not ObjectNode`() {
57 + fun `map should return null when result is not ObjectNode and
    caseDefinitionVersionTag is missing`() {
58 58          val event = ValtimoEvent(
59 59              id = "test-id",
60 60              type = ConfigurationIssueUpdated.TYPE,
  @@ -68,10 +68,7 @@ class ConfigurationIssueSseEventManagerTest {
68 68
69 69          val result = mapper.map(event)
70 70
71 -
71 +      assertThat(result).isNull()
72 -      val sseEvent = result as ConfigurationIssueUpdatedSseEvent
73 -      assertThat(sseEvent.caseDefinitionKey).isEqualTo("fallback-key")
74 -      assertThat(sseEvent.caseDefinitionVersionTag).isEqualTo("")
75 72      }
76 73
77 74      @Test
  
```

```

  ...cument/mapper/DocumentSseEventManagerTest.kt
  @@ -82,6 +82,16 @@ class DocumentSseEventManagerTest {
82 82      assertEquals(documentId, (sseEvent as
    DocumentUpdatedSseEvent).documentId)
83 83      }
84 84
85 +      @Test
86 +      fun `should return null when document updated event has no resultId`() {
87 +          val valtimoEvent = mock<ValtimoEvent>()
88 +          whenever(valtimoEvent.type).thenReturn(DocumentUpdated.TYPE)
89 +          whenever(valtimoEvent.resultId).thenReturn(null)
90 +
  
```

```

91 +         val sseEvent = documentSseEventManager.map(valtimoEvent)
92 +         assertNull(sseEvent)
93 +     }
94 +
85 95     @Test
86 96     fun `should map to null with unknown type event`() {
87 97         val valtimoEvent = mock<ValtimoEvent>()

```



▼ ...n/com/ritense/inbox/InboxHandlingService.kt



```

↑... @@ -22,7 +22,7 @@ class InboxHandlingService(
22 22     private val eventHandlers: List<InboxEventHandler>,
23 23     ) {
24 24     fun handle(message: String) {
25 25 -         logger.info("Received message: {}", message)
25 25 +         logger.debug { "Received inbox message" }
26 26         eventHandlers.forEach { it.handle(message) }
27 27     }
28 28

```



▼ ...om/ritense/inbox/ValtimoCloudEventManager.kt



```

↑... @@ -21,6 +21,7 @@ import com.fasterxml.jackson.module.kotlin.readValue
21 21     import io.cloudevents.CloudEvent
22 22     import io.cloudevents.core.provider.EventFormatProvider
23 23     import io.cloudevents.jackson.JsonFormat
24 24 + import io.github.oshai.kotlinlogging.KotlinLogging
24 25
25 26     class ValtimoCloudEventManager(
26 27         private val objectMapper: ObjectMapper
↑... @@ -34,6 +35,7 @@ class ValtimoCloudEventManager(
34 35         return try {
35 36             cloudEventFormat.deserialize(payload.encodeToByteArray())
36 37         } catch (ex: Exception) {
38 38 +             logger.warn(ex) { "Failed to deserialize CloudEvent from payload" }
37 39             null
38 40         }
39 41     }

```



```

↑... @@ -52,7 +54,12 @@ class ValtimoCloudEventManager(

```

```

52 54         result = cloudEventData?.result
53 55     )
54 56     } catch (ex: Exception) {
57 +         logger.warn(ex) { "Failed to map CloudEvent to ValtimoEvent" }
55 58         null
56 59     }
57 60 }

61 +
62 +     companion object {
63 +         private val logger = KotlinLogging.logger {}
64 +     }

58 65 }

```

▼ ...e/outbox/rabbitmq/RabbitMessagePublisher.kt

```

↑... @@ -40,7 +40,7 @@ class RabbitMessagePublisher(
40 40     rabbitTemplate.exchange ?: ""
41 41     }
42 42     private val routingKey: String = routingKey ?: run {
43 -         logger.debug { "Using Rabbit template default routingKey:
    ${rabbitTemplate.exchange}" }
43 +         logger.debug { "Using Rabbit template default routingKey:
    ${rabbitTemplate.routingKey}" }
44 44     rabbitTemplate.routingKey ?: ""
45 45     }
46 46

↕... @@ -58,14 +58,15 @@ class RabbitMessagePublisher(
58 58
59 59     try {
60 60         val result = correlationData.future[deliveryTimeout.toMillis(),
    TimeUnit.MILLISECONDS]
61 -         if (!result!!.isAck) {
62 -             throw MessagePublishingFailed("Outbox message was not
    acknowledged: reason=${result.reason}, routingKey=${routingKey},
    msgId=${message.id}, correlationId= ${correlationData.id}\")
63 -         } else if (correlationData.returned != null) {
64 -             val returned = correlationData.returned!!
65 -             throw MessagePublishingFailed("Could not deliver outbox message:
    routingKey=${returned.routingKey}, code=${returned.replyCode},
    msg=${returned.replyText}, routingKey=${routingKey}, msgId=${message.id},
    correlationId= ${correlationData.id}\")

```

```

61 +         ?: throw MessagePublishingFailed("Outbox message confirmation
    result was null: routingKey=${routingKey}, msgId=${message.id},
    correlationId=${correlationData.id}")
62 +         if (!result.isAck) {
63 +             throw MessagePublishingFailed("Outbox message was not
    acknowledged: reason=${result.reason}, routingKey=${routingKey},
    msgId=${message.id}, correlationId=${correlationData.id}")
64 +         }
65 +         correlationData.returned?.let { returned ->
66 +             throw MessagePublishingFailed("Could not deliver outbox message:
    routingKey=${returned.routingKey}, code=${returned.replyCode},
    msg=${returned.replyText}, routingKey=${routingKey}, msgId=${message.id},
    correlationId=${correlationData.id}")
66 67         }
67 68     } catch (timeoutException: TimeoutException) {
68 -         throw MessagePublishingFailed("Outbox message delivery was not
    confirmed in time: routingKey=${routingKey}, msgId=${message.id}, correlationId=
    ${correlationData.id}")
69 +         throw MessagePublishingFailed("Outbox message delivery was not
    confirmed in time: routingKey=${routingKey}, msgId=${message.id},
    correlationId=${correlationData.id}")
69 70     }
70 71 }
71 72

```

...tion/release-notes/13.x.x/13.22.0/README.md

```

@@ -18,4 +18,4 @@
18 18
19 19     ## Bugfixes
20 20
21 - * New bugfix.
21 + * Fixed sensitive data logging in inbox messages, silent exception swallowing in
    cloud event mapping, and null safety issues in SSE event mappers.

```

Comments 0



Please [sign in](#) to comment.