

Diagnostic Analysis Report

Customer Problem

initial_thread_dump_analysis

Log Analysis

Summary of Findings:

The logs display a regular pattern of successful log messages stating "Welcome to HealthcareService" interleaved with a single error event. At 14:34:19, an error occurs during JSON payload transformation when attempting to parse an element labeled jsonObject. The error message indicates an illegal character "<d>" encountered during the JSON-to-XML conversion process. The stack trace reveals that the issue surfaced within components responsible for handling JSON payloads, particularly during the creation and consumption of JSON streams. This suggests that malformed JSON or unexpected characters in the payload could be causing parser failures.

Suspected components include areas that perform JSON parsing and transformation, such as the JSON stream reader, utility classes for JSON manipulation, and the payload factory mediator that utilizes these utilities. The timing of the error amidst successful log message processing may indicate that only specific payloads or conditions trigger the faulty behavior.

```
SUSPECTED_CLASSES: [{ 'package': 'org.apache.synapse.commons.staxon.core.base',  
'class': 'AbstractXMLStreamReader', 'issue_line': 245}, { 'package':  
'org.apache.synapse.commons.staxon.core.json', 'class': 'JsonXMLStreamReader',  
'issue_line': 88}, { 'package': 'org.apache.synapse.commons.json', 'class': 'JsonUtil',  
'issue_line': 942}, { 'package': 'org.apache.synapse.commons.json', 'class':  
'JsonDataSource', 'issue_line': 154}, { 'package':  
'org.apache.synapse.mediators.transform', 'class': 'PayloadFactoryMediator',  
'issue_line': 145}]
```

```
ERROR_MESSAGE: "javax.xml.stream.XMLStreamException: java.io.IOException: Illegal  
character: <d>"
```

Comprehensive Thread Analysis

Not applicable - no problematic threads identified.

Conclusions and Recommendations

Diagnosis: The error is caused by the presence of an illegal character ("`<d>`") within the JSON payload during its transformation to XML.

Diagnosis: The parser fails when processing malformed JSON or payloads that include unexpected data, likely originating from specific payloads that do not comply with the expected format.

Diagnosis: The issue implicates components involved in JSON reading and transformation (e.g., `JSONStreamReader`, `JsonUtil`, and `PayloadFactoryMediator`) where the error was thrown at specific lines in these classes.

Actionable Steps:

Actionable Step: Validate and sanitize all incoming JSON payloads before they are processed by transformation components.

Actionable Step: Implement stricter error handling and logging around the JSON-to-XML conversion process to capture unexpected characters early.

Actionable Step: Review and update the parsing logic in the implicated classes (`AbstractXMLStreamReader`, `JsonXMLStreamReader`, `JsonUtil`, `JsonDataSource`, and `PayloadFactoryMediator`) to better handle or reject malformed JSON inputs.

Actionable Step: If applicable, apply input validation at the API gateway or data ingestion layer to ensure that only well-formed JSON is accepted.

Actionable Step: Conduct unit and integration tests focused on edge cases with malformed payloads to ensure the improved error handling resolves the issue.