# Azure API Management Inbound Policy Documentation

## Client Lookup Service - Infor LN Integration

### Document Information

* **Document Version**: 1.0
* **Created Date**: August 29, 2025
* **Service Name**: Client Lookup Service
* **Backend System**: Infor LN
* **Policy Type**: Azure API Management Inbound Policy

## Executive Summary

This document provides comprehensive documentation for an Azure API Management (APIM) inbound policy that facilitates client lookup operations with the Infor LN system. The policy implements robust error handling, request validation, correlation tracking, and standardized cXML response formatting.

## Service Overview

### Purpose

The Client Lookup Service acts as a gateway between external clients and the Infor LN system, providing: - Client configuration validation - Request correlation tracking  
- Standardized error responses in cXML format - Parameter validation and routing

### Architecture Components

* **Frontend**: Azure API Management Gateway
* **Backend**: Infor LN System ({{INFOR-LN-BASE-URL}})
* **Protocol**: HTTP REST API
* **Response Format**: cXML 1.2.014

## Policy Structure Analysis

### 1. Inbound Processing Section

#### 1.1 Correlation ID Generation

<set-variable name="correlationId" value="@{  
 var existingId = context.Request.Headers.GetValueOrDefault("X-Correlation-Id", "");  
 return !string.IsNullOrEmpty(existingId) ? existingId : Guid.NewGuid().ToString();  
}" />

**Functionality:** - Checks for existing X-Correlation-Id header - Generates new GUID if not present - Ensures request traceability across systems

#### 1.2 Parameter Extraction

<set-variable name="fromHeader" value="@{  
 return context.Request.Url.Query.GetValueOrDefault("fromHeader");  
}" />  
<set-variable name="toHeader" value="@{  
 return context.Request.Url.Query.GetValueOrDefault("toHeader");  
}" />

**Required Parameters:** - fromHeader: Source client identifier - toHeader: Destination client identifier

#### 1.3 Parameter Validation

The policy implements strict validation ensuring both parameters are present and non-empty:

**Validation Logic:** - Checks if parameters are null, empty, or whitespace - Returns HTTP 400 with cXML error response if validation fails

**Error Response Format:**

<?xml version="1.0" encoding="UTF-8"?>  
<cXML version="1.2.014" payloadID="{guid}@apim-gateway" timestamp="{utc-timestamp}">  
 <Response>  
 <Status code="400" text="Bad Request">  
 Missing required query parameters: fromHeader and toHeader are both mandatory  
 </Status>  
 </Response>  
</cXML>

#### 1.4 Backend Configuration

* **Base URL**: {{INFOR-LN-BASE-URL}} (configurable variable)
* **Endpoint**: /api/client/lookup
* **Method**: Preserves original HTTP method
* **Parameters**: Forwards validated fromHeader and toHeader

### 2. Backend Processing Section

The policy uses simple request forwarding:

<backend>  
 <forward-request />  
</backend>

**Behavior:** - Forwards the modified request to Infor LN system - Maintains original request body and headers - Adds correlation tracking headers

### 3. Outbound Processing Section

#### 3.1 HTTP 404 Handling (Client Not Found)

<when condition="@(context.Response.StatusCode == 404)">

**Response:** - **HTTP Status**: 404 Not Found - **Content-Type**: application/xml - **Payload ID**: {guid}@infor-ln - **Message**: Specific client configuration not found

#### 3.2 HTTP 4xx Client Errors

<when condition="@(context.Response.StatusCode >= 400 && context.Response.StatusCode < 500)">

**Handles:** - 400 Bad Request - 401 Unauthorized  
- 403 Forbidden - 405 Method Not Allowed - Other 4xx errors

**Response Format:** - Preserves original status code and reason - Wraps in cXML format with descriptive message

### 4. Error Handling Section

#### 4.1 Gateway Timeout (504)

**Trigger Conditions:** - Request timeout to backend system - Infor LN system unresponsive

**Response:**

<Status code="504" text="Gateway Timeout">  
 Infor LN system did not respond in time  
</Status>

#### 4.2 Bad Gateway (502)

**Trigger Conditions:** - Connection refused - DNS resolution failure - Network connectivity issues

**Response:**

<Status code="502" text="Bad Gateway">  
 Unable to connect to Infor LN system  
</Status>

#### 4.3 Internal Server Error (500)

**Trigger Conditions:** - Unexpected policy execution errors - System exceptions

**Response:**

<Status code="500" text="Internal Server Error">  
 An unexpected error occurred while processing the request  
</Status>

## Request/Response Flow

### Successful Request Flow

1. **Client Request** → APIM Gateway
2. **Correlation ID** → Generated/Extracted
3. **Parameter Validation** → fromHeader & toHeader validated
4. **Request Transformation** → Route to /api/client/lookup
5. **Backend Call** → Forward to Infor LN
6. **Successful Response** → Return original response

### Error Request Flow

1. **Client Request** → APIM Gateway
2. **Validation Failure** → Return cXML error response
3. **Backend Error** → Transform to cXML format
4. **System Error** → Generate appropriate cXML error

## Configuration Requirements

### Environment Variables

* {{INFOR-LN-BASE-URL}}: Base URL for Infor LN system

### Required Headers

* X-Correlation-Id (optional - auto-generated if missing)
* X-APIM-Correlation-Id (added by policy for backend tracing)

### Query Parameters

* fromHeader (mandatory): Source client identifier
* toHeader (mandatory): Destination client identifier

## Error Codes and Messages

| HTTP Code | Scenario | cXML Status | Description |
| --- | --- | --- | --- |
| 400 | Missing Parameters | Bad Request | Required query parameters missing |
| 404 | Client Not Found | Client Not Found | No configuration found for specified headers |
| 4xx | Client Errors | Varies | Business validation failures |
| 500 | System Error | Internal Server Error | Unexpected policy/system errors |
| 502 | Connectivity | Bad Gateway | Cannot connect to Infor LN |
| 504 | Timeout | Gateway Timeout | Infor LN system timeout |

## Monitoring and Troubleshooting

### Key Metrics to Monitor

* **Request Volume**: Number of client lookup requests
* **Error Rates**: 4xx and 5xx response percentages
* **Response Times**: Latency to Infor LN system
* **Timeout Frequency**: 504 error occurrences

### Troubleshooting Guide

#### High 400 Error Rates

* **Root Cause**: Missing or invalid query parameters
* **Solution**: Validate client integration and parameter passing

#### High 404 Error Rates

* **Root Cause**: Client configurations not found in Infor LN
* **Solution**: Verify client setup in Infor LN system

#### High 502/504 Error Rates

* **Root Cause**: Infor LN connectivity or performance issues
* **Solution**: Check network connectivity and Infor LN system health

### Correlation Tracking

* Use X-Correlation-Id header to trace requests across systems
* Each response includes unique payloadID for identification
* Correlation IDs are preserved throughout the request lifecycle

## Security Considerations

### Input Validation

* Strict parameter validation prevents injection attacks
* Query parameter sanitization implemented
* Error messages don’t expose internal system details

### Error Information Disclosure

* Generic error messages prevent system information leakage
* Internal error details not exposed to external clients
* Correlation IDs enable internal debugging without security risk

## Best Practices Implementation

### 1. Correlation Tracking

* Implements distributed tracing pattern
* Enables end-to-end request tracking
* Facilitates debugging and monitoring

### 2. Graceful Error Handling

* All error scenarios handled with appropriate HTTP status codes
* Consistent cXML response format maintained
* Descriptive but secure error messages

### 3. Configuration Management

* Environment-specific backend URLs through variables
* Separation of configuration from policy logic
* Easy deployment across environments

### 4. Standards Compliance

* cXML 1.2.014 format compliance
* Proper HTTP status code usage
* RESTful API design principles

## Maintenance and Updates

### Regular Maintenance Tasks

* Monitor error rates and response times
* Review and update timeout configurations
* Validate Infor LN system connectivity
* Update cXML version compatibility as needed

### Policy Updates

* Version control all policy changes
* Test in development environment first
* Maintain backward compatibility
* Document all modifications

## Conclusion

This Azure API Management inbound policy provides a robust, secure, and maintainable solution for client lookup operations with the Infor LN system. The implementation follows industry best practices for API gateway patterns, error handling, and monitoring, ensuring reliable service operation and easy maintenance.

The policy’s comprehensive error handling, correlation tracking, and standardized response formatting make it suitable for production enterprise environments requiring high reliability and observability.