Software Requirements Specification (SRS)

Project Title: ERP REST API Enhancement for Contract Handling

Version: 1.0  
Date: 2025-08-31

1. Introduction

1.1 Purpose

This document specifies the software requirements for enhancing an existing Java API to support REST requests from an ERP system where the contract number parameter can be NULL. The enhancement involves decoupling the HTTP response from downstream processing to address timeout issues when handling multiple contracts.

1.2 Scope

Currently, the API supports a request with a single contract number as mandatory field and performs downstream processing via ERP API within the HTTP request/response cycle. This enhancement enables handling of requests where the contract number is NULL, triggering internal logic to identify multiple eligible contracts and process them asynchronously.

1.3 Definitions, Acronyms, and Abbreviations

| Term | Description |
| --- | --- |
| ERP | Enterprise Resource Planning |
| API | Application Programming Interface |
| REST | Representational State Transfer |
| HTTP | HyperText Transfer Protocol |

2. Overall Description

2.1 Product Perspective

The product is an enhancement of an existing Java-based REST API that interfaces with an ERP system. The API currently supports synchronous processing for a single contract.

2.2 User Needs

Users (ERP system clients) need the ability to:

* Submit requests with or without a contract number.
* Receive a timely acknowledgment response regardless of the number of contracts processed.
* Ensure downstream processes are completed reliably even when processed asynchronously.

2.3 Assumptions and Dependencies

* The database contains accurate and up-to-date information to identify eligible contracts.
* ERP downstream API is capable of handling multiple sequential calls.
* Background processing infrastructure is available and scalable.

3. System Features and Requirements

3.1 AS-IS Functionality

* REST API accepts a request with 5 parameters, including a required contractNumber.

|  |
| --- |
| grpcurli --dv-auth SELF -d  '{ "value":  { "offerName": "AXIOM",  "productName": "",  "contractNumber": "AXL12345",  "execDate": "01-JAN-2025",  "runMode": "FINAL" }  }' localhost:9999 proto.com.axiom.findotrevenuebackend.FinRevenueBackendsApi/processRevRec |
|  |

* Calls the ERP API for downstream processing synchronously.
* Returns the ERP API response in the HTTP response.
* HTTP timeout is 120 seconds, sufficient for a single contract.

3.2 TO-BE Functionality

New Use Case: contractNumber = NULL

1. The API shall accept REST requests where the contractNumber parameter can be NULL.

|  |
| --- |
| grpcurli --dv-auth SELF -d  '{ "value":  { "offerName": "AXIOM",  "productName": "",  "contractNumber": "AXL12345",  "execDate": "01-JAN-2025",  "runMode": "FINAL" }  }' localhost:9999 proto.com.axiom.findotrevenuebackend.FinRevenueBackendsApi/processRevRec |
|  |

1. When contractNumber is NULL, the API will query the database using the remaining four parameters to identify all eligible contracts.
2. Upon receiving such a request, the API will immediately return an acknowledgment response: "Request received and processing in progress".
3. The processing for eligible contracts will happen asynchronously in the background.
4. For each eligible contract, the background process will call the standard ERP API individually.
5. Any failure during processing of a specific contract will be logged without affecting the processing of others.
6. Optionally, the system may provide a way to track the status of background processing per request (e.g., via a future enhancement).

3.3 Non-Functional Requirements

1. The HTTP response for a request with NULL contract number must be returned within 10 seconds to avoid client timeout.
2. Processing logs must be retained and accessible for audit and troubleshooting purposes.
3. The system must ensure idempotency so that retrying the same request does not result in duplicate downstream processing.
4. Background jobs should be resilient to transient failures and support retry mechanisms where applicable.

4. System Architecture Overview (High-Level)

1. Request Received
   * Accept REST request with or without contract number.
2. Routing Logic
   * If contractNumber != NULL: Proceed with synchronous ERP API call.
   * If contractNumber == NULL:
     + Query eligible contracts using SQL.
     + Trigger background processing task.
3. Background Worker
   * Loop through contracts and invoke ERP API per contract.
   * Log outcomes per contract.

A diagram of a process

AI-generated content may be incorrect.

5. Constraints

* Must maintain backward compatibility for existing clients.
* Processing must not exceed API rate limits imposed by ERP system.
* Background processing infrastructure must be scalable.