Test Plan for SOAP API – Number Conversion

Created by: Ankit Kumar

# 1. Objective

This document outlines the test plan for the SOAP API – Number Conversion service.   
The objective is to ensure that all features and functionalities of the API work as expected for the target audience, i.e., developers, QA engineers, and system integrators who consume the SOAP-based Number Conversion service.

# 2. Scope

Features to be tested:  
- Conversion of numbers into words (NumberToWords SOAP method).  
- Handling of valid, invalid, and edge-case inputs.  
- Response validation for different scenarios (positive and negative).  
  
Types of testing:  
- Manual testing (via Postman).  
- Automated testing (via Newman / CI integration).  
- Negative testing (missing headers, body, invalid input).  
- Performance testing (basic response time checks).  
  
Environments:  
- HTTPS: https://www.dataaccess.com/webservicesserver/NumberConversion.wso  
- HTTP (negative scenario): http://www.dataaccess.com/webservicesserver/NumberConversion.wso  
  
Evaluation criteria:  
- Number of defects found.  
- Time taken to complete testing.  
- API compliance with SOAP standards.  
- Correctness of conversion results.  
  
Team roles and responsibilities:  
- Test Lead: Review & approve test cases, monitor progress.  
- Testers: Execute test cases, log defects.  
- Developers: Fix defects and provide technical clarifications.  
- Stakeholders: Review reports and approve releases.

# 3. Inclusions

Introduction: This test plan validates the SOAP API service’s ability to convert numeric input to word format reliably.  
  
Test Objectives:  
- Validate correctness of conversions.  
- Identify defects and integration issues.  
- Ensure the service handles invalid/negative scenarios gracefully.  
- Ensure SOAP compliance for headers, body, and response structure.

# 4. Exclusions

- No load testing at enterprise scale.  
- No UI testing (service-level only).  
- No deep security penetration testing.

# 5. Test Environments

Operating Systems: Windows 10, Linux (for Newman CLI)  
Browsers: N/A (SOAP API testing via Postman)  
Devices: Desktop/Laptop only  
Network Connectivity: Wi-Fi / wired network  
Hardware/Software Requirements: 8 GB RAM, i5 processor, Postman installed  
Security Protocols: HTTPS supported  
Access Permissions: Open public API (no authentication)

# 6. Defect Reporting Procedure

Criteria for identifying defects:  
- Incorrect conversion output  
- Missing/invalid response structure  
- Server errors (5xx, 4xx)  
  
Steps for reporting defects:  
- Log defect in JIRA with request details, response details, steps to reproduce, and screenshots/logs.  
  
Triage and prioritization:  
- Severity: Critical, High, Medium, Low  
- Priority: P1 (immediate fix), P2, P3  
  
Tracking tools: JIRA / Azure DevOps  
Roles and responsibilities: Testers log defects, Developers fix and update status, Test Lead ensures proper closure  
Communication channels: Daily stand-ups, defect triage meetings  
Metrics: Defect density, time to fix, reopen rate

# 7. Test Strategy

Step 1: Test scenarios and test cases creation  
- Techniques: Equivalence Partitioning, Boundary Value Analysis, Error Guessing, Negative testing  
  
Step 2: Testing procedure  
- Smoke Testing: Validate endpoint availability  
- Functional Testing: Validate correctness of responses  
- Regression Testing: Re-run all scenarios after bug fixes  
- Exploratory Testing: Try invalid/edge cases  
  
Step 3: Best Practices  
- Shift-left testing with early validation  
- End-to-end flow validation  
- Parallel execution using Newman

# 8. Test Schedule

- Test Plan Creation: Day 1  
- Test Case Design: Day 2  
- Test Execution: Day 3 – Day 4  
- Defect Reporting & Retesting: Day 5 – Day 6  
- Test Closure & Report: Day 7

# 9. Test Deliverables

- Test Scenarios & Test Cases  
- Test Data  
- Execution Results  
- Defect Logs  
- Test Summary Report

# 10. Entry and Exit Criteria

Requirement Analysis  
- Entry: Postman collection available  
- Exit: Requirements understood, scenarios prepared  
  
Test Execution  
- Entry: Signed-off test cases, stable API endpoint  
- Exit: Test execution completed, defects logged  
  
Test Closure  
- Entry: Test case results, defect reports ready  
- Exit: Final Test Summary Report prepared

# 11. Tools

- Postman (manual execution)  
- Newman CLI (automation)  
- JIRA (defect tracking)  
- Excel/Word (test documentation)

# 12. Risks and Mitigations

- Risk: Service downtime -> Mitigation: Retry later / run against staging  
- Risk: Incorrect schema updates -> Mitigation: Validate with WSDL  
- Risk: Limited test data -> Mitigation: Generate dynamic test inputs

# 13. Approvals

Documents for Client Approval:  
- Test Plan  
- Test Scenarios  
- Test Cases  
- Test Summary Report