# Test Plan for Number Conversion SOAP API

Created by: Sandhya Sankeshwar

## 1. Objective

This document outlines the test plan for the Number Conversion SOAP API. The objective is to ensure that all features and functionalities work as expected for API consumers who require number-to-word conversion services.

## 2. Scope

The scope of this test plan includes:

1. **Features to be tested:** 
   * Processing numeric input and returning word representation.
   * Handling various numeric ranges and invalid inputs.
   * Response format validation (XML structure).
   * Error handling and response codes.
2. **Types of testing:**

* Manual Testing
* Automated testing with specific frameworks (e.g., Postman Newman, SoapUI Pro)
* Performance testing with load and stress scenarios
* Security testing including XML injection vulnerability assessment
* Compatibility testing across different SOAP clients

1. Evaluation criteria:
   * Number of defects found.
   * Performance metrics:

- Average response time (target: <500ms)

- Requests per second (RPS)

- Error rate percentage

* + Compliance with API specifications.
  + Test coverage percentage

1. Team roles and responsibilities:
   * Test Lead:

- Oversees testing activities

- Develops test strategy

- Manages test resource allocation

* + **Senior Tester:**

**-** Develops complex test scenarios

**-** Performs exploratory testing

**-** Designs automation scripts

* + Testers:

- Execute test cases and report defects

- Report and track defects

- Perform regression testing

* + Developers:

- Fix reported issues.

- Provide clarifications on requirements

- Assist in reproducing complex scenarios

* + Stakeholders:

- Review test results.

- Provide final sign-off

- Approve release decisions

## 3. Test Strategy

Step 1: Test Scenarios and Test Case Creation

* **Techniques:**
  + Equivalence Class Partitioning
  + Boundary Value Analysis
  + Error Guessing
  + Exploratory Testing

#### Step 2: Testing Procedure

* **Smoke Testing:** Verify API is responsive.
* **Functional Testing**: Validate expected number-to-word conversions.
* **Negative Testing:** Provide invalid inputs to check error handling.
* **Performance Testing:** Validate response time under load.

## 4. Test Environments

* **Operating Systems:** Windows 10, macOS, Linux.
* **Browsers:** Google Chrome, Mozilla Firefox, Microsoft Edge (for API testing tools compatibility).
* **Network Connectivity:** Wi-Fi, Cellular connections, Wired enterprise network
* **Security Protocols:** Authentication mechanisms, if applicable.
* **Access Permissions:** Tester and developer roles.

## 5. Defect Reporting Procedure

• **Criteria for defects:** Deviation from expected output, incorrect error handling, unexpected API failures.

• **Reporting steps:** Use a designated template, provide detailed reproduction steps, attach logs/screenshots.

• **Triage & Prioritization:**

- **Critical:** API fails to return a response.

- **High:** Incorrect conversion of numbers.

- **Medium**: Performance issues.

**- Low:** Minor inconsistencies.

## 6. Test Schedule

• **Test Plan Preparation:** [Start Date] - [End Date]

• **Test Case Development**: [Start Date] - [End Date]

• **Test Execution:** [Start Date] - [End Date]

• **Defect Reporting & Fixing:** Continuous

• **Test Closure:** [End Date]

## 7. Test Deliverables

• Test Plan Document

• Test Cases (Maintained separately)

• Defect Reports

• Test Summary Report

## 8. Entry and Exit Criteria

### 8.1 Requirement Analysis

• **Entry:** Requirement documents available.

• **Exit:** Requirements clarified and documented.

### 8.2 Test Execution

• **Entry:** Signed-off test cases and test data.

• **Exit:** Test case execution completed, defect reports available.

### 8.3 Test Closure

• **Entry:** Defect fixes verified.

• **Exit:** Test summary report generated.

## 9. Tools

• JIRA (Defect Tracking)

• Postman/SOAP UI (API Testing)

• Snipping Tool (Screenshots)

• Excel/Word (Test Documentation)

## 10. Risks and Mitigations

• Risk: API endpoint downtime. Mitigation: Use mock services for testing.

• Risk: Late requirement changes. Mitigation: Agile approach with continuous updates.

• Risk: Inconsistent test environments. Mitigation: Standardize test setup.

## 11. Approvals

Team will send different types of documents for Client Approval like below:

• Documents for Approval:

- Test Plan

- Test Scenarios & Cases

- Test Summary Report

Testing will only continue to the next steps once these approvals are done