Test Plan for SOAP Request Project:

Number of words

**Created by:** Anuj Rajput  
 **Project Start Date:** 1 May 2025  
 **Project Duration:** 2 Weeks  
 **Team:** 2 QA Resources (1 QA Lead, 1 QA Engineer)

## **1. Objective**

This document outlines the test plan for the **Number to Words SOAP Web Service**. The objective is to ensure that the service returns the correct English word representation of a positive number (up to quadrillions) when a valid SOAP request is made using the following endpoint:

**URL:**<https://www.dataaccess.com/webservicesserver/NumberConversion.wso?op=NumberToWords>

## **2. Scope**

### **Features to be Tested:**

* Valid number-to-word conversion.
* Handling of edge cases (0, maximum value, etc.).
* Response schema validation.
* SOAP envelope structure and headers.
* Error handling (negative numbers, invalid inputs).

### **Types of Testing:**

* Functional Testing (manual)
* API Automation Testing
* Performance Testing (limited)
* Compatibility Testing (different SOAP clients/tools)

### **Environments:**

* SOAP UI
* Postman (with SOAP support)
* Custom-built Java/Python clients (optional)

### **Evaluation Criteria:**

* 100% test case coverage for defined scenarios
* All critical bugs resolved before closure
* Response correctness and latency

### **Team Roles:**

* **QA Lead:** Oversees planning, reviews, reports.
* **QA Engineer:** Creates and executes test cases, reports defects.

## **3. Inclusions**

* Introduction and objective of testing.
* End-to-end validation of the NumberToWords operation.
* Verification of SOAP envelope and body structure.
* Test data variation covering typical and edge input cases.

## **4. Exclusions**

* Other operations from the same web service (like NumberToDollars).
* UI-level integration (since this is a backend SOAP service).
* Non-English number conversion (only English is supported).

## **5. Test Environments**

* **Operating Systems:** Windows 10, Linux
* **SOAP Tools:** SOAP UI, Postman
* **Network:** Standard broadband
* **Hardware:** 4 GB+ RAM, modern CPU
* **Security:** Public test API – no authentication
* **Access:** QA Lead and QA Engineer have full test access

## **6. Defect Reporting Procedure**

* Use: **JIRA**
* Defect Format: Summary, Steps, Expected vs Actual, Severity
* Priority & Severity Matrix
* QA Lead to triage and assign
* Daily bug triage meetings (if needed)
* Metrics: Total bugs, resolved, severity distribution

## **7. Test Strategy**

### **Step 1: Test Case Design**

* Techniques: Boundary Value Analysis, Equivalence Class Partitioning
* Use Case Testing: User calls SOAP request with different numbers
* Negative Testing: Sending invalid/negative values

### **Step 2: Testing Procedure**

* **Smoke Testing:** Basic connectivity and response
* **Functional Testing:** Validate output word for number
* **Regression Testing:** Validate no impact from changes
* **UI Testing (limited):** Not applicable, service-only testing

### **Step 3: Best Practices**

* Shift Left Testing: Early involvement in test design
* Exploratory Testing: Additional edge scenarios
* End-to-End: Complete request-response cycle validation

## **8. Test Schedule**

| **Task** | **Duration** | **Responsible** |
| --- | --- | --- |
| Test Plan Creation | 1 day | QA Lead |
| Test Case Design | 2 days | QA Engineer |
| Test Execution | 6 days | Both |
| Defect Reporting & Fixes | Continuous | QA Engineer + Dev |
| Test Closure & Reporting | 2 days | QA Lead |

## **9. Test Deliverables**

* Test Plan
* Test Case Document
* Test Summary Report
* Defect Reports

## **10. Entry and Exit Criteria**

### **Requirement Analysis:**

* **Entry:** Access to WSDL and SOAP operation description
* **Exit:** Clear understanding of service behavior

### **Test Execution:**

* **Entry:** Approved test cases and stable endpoint
* **Exit:** All test cases executed, critical defects closed

### **Test Closure:**

* **Entry:** Final test execution report ready
* **Exit:** Test Summary Report delivered

## **11. Tools**

* **JIRA:** Defect tracking
* **SOAP UI/Postman:** API Testing
* **Excel/Word:** Documentation
* **Snipping Tool:** Screenshot capture

## **12. Risks and Mitigations**

| **Risk** | **Mitigation** |
| --- | --- |
| Resource Unavailability | Cross-training between QA Lead and Engineer |
| Endpoint Downtime | Use mock services to simulate |
| Short Timeline | Prioritize critical test cases first |

## **13. Approvals**

* Documents to be reviewed and approved by QA Lead and Project Manager:  
  + Test Plan
  + Test Scenarios
  + Test Summary Report