

Test Plan: IsValidISBN13 SOAP API

Objective

The goal of this test plan is to ensure the quality, functionality, and reliability of the **IsValidISBN13** SOAP API hosted at <http://webservices.daehosting.com/services/isbnservice.wso>.

The API is designed to validate whether a given ISBN-13 number is correct by returning a boolean response.

Scope

Scope of Test Plan for **IsValidISBN13** SOAP API:

1. **Functional Testing**
 - Validate the core ISBN validation logic with valid and invalid inputs.
 - Ensure the API behaves correctly with boundary values, empty or malformed data.
2. **Data Validation Testing**
 - Validate handling of numerical and non-numerical data.
 - Check minimum and maximum length of the input field.
3. **Error Handling Testing**
 - Validate error responses for malformed SOAP envelopes.
 - Ensure proper error messages are returned for invalid request formats.
4. **Performance Testing**
 - Assess response time under normal and peak loads.
 - Measure throughput and scalability.
5. **Security Testing**
 - Check for vulnerabilities like SOAP injection.
 - Ensure no sensitive data is leaked in response.
6. **Integration Testing**
 - Test if the SOAP service integrates well with client systems.
7. **Compatibility Testing**
 - Test across different environments (Windows, Linux, etc.).
8. **Documentation Review**
 - Ensure API documentation aligns with actual service behavior.
9. **Load Testing**
 - Simulate concurrent users to validate stability.
10. **Regression Testing**
 - Re-run functional scenarios after code updates.
11. **Edge Case Testing**

- Use extremely long, short, or special-character strings.
 - 12. **Usability Testing**
 - Evaluate the SOAP interface usability from the client developer perspective.
 - 13. **CI/CD Testing**
 - Ensure automation scripts work in deployment pipelines.
 - 14. **Rate Limiting Testing**
 - Check if API handles excessive requests correctly.
-

Inclusions

- Valid ISBN-13 Check: 9780306406157
 - Invalid ISBN-13 Check: 9780306406158
 - Empty string input
 - Special characters in ISBN
 - Whitespace-trimmed inputs
 - Inputs exceeding 13 characters or less than 13
 - Non-numeric input: ABCDEFGHIJKLM
 - Malformed XML in SOAP request
 - High volume request simulation
-

Test Environments

OS/Platform	Browsers/Tools
Windows 10	Postman, SOAP UI, Chrome
macOS	Safari, Terminal (curl)
Android/iOS	Mobile clients via apps
Linux	curl, Python scripts

Security/Access: Open endpoint, no authentication required.

Defect Reporting Procedure

- Tool: JIRA
 - Include: Request payload, response, timestamps, environment, and screenshots.
 - Prioritize by severity and reproducibility.
 - Defects reviewed by Test Lead and assigned accordingly.
-

Test Strategy

- **Design Techniques:**
 - Equivalence Partitioning
 - Boundary Value Analysis
 - Error Guessing
 - Use Case Testing
- **Testing Types:**
 - Smoke, Sanity, Functional, Regression
 - Exploratory, Load, Integration, Security
- **Execution Flow:**
 - Smoke test → Functional → Load & Negative Testing → Regression → Closure

Test Schedule

Task	Timeline
Test Plan Creation	Day 1
Test Case Creation	Day 1-2
Test Execution	Day 3-5
Summary Report	Day 6

Sprint-based testing with 1-week iteration.

Test Deliverables

- Test Plan Document
- Test Case Document
- Execution Report
- Defect Logs
- Final Summary Report

Entry and Exit Criteria

Requirement Analysis

- Entry: API specs available
- Exit: Functional understanding confirmed

Test Execution

- Entry: Test cases approved, stable environment
- Exit: All planned test cases executed, defects logged

Test Closure

- Entry: Execution and defect reports ready
- Exit: Summary reports shared with stakeholders

Tools

- JIRA (Defect Tracking)
- SOAP UI/Postman (Test Execution)
- Python Scripts (Automation)
- Word/Excel (Documentation)

Risks and Mitigations

Risk	Mitigation
Endpoint downtime	Notify dev team, test on staging
Unexpected input format behavior	Add comprehensive validation tests
Tight timelines	Prioritize high-risk tests first

Approvals

- This Test Plan, once reviewed and approved by the QA Lead and stakeholders, will guide the test execution.
- Sign-off required before proceeding to execution phase.