**TEST PLAN – SOAP API: Temperature Conversion**

**1. Project Overview**

This test plan covers the functional, boundary, and negative test scenarios for the SOAP web service endpoint CelsiusToFahrenheit offered by [W3Schools TempConvert](https://www.w3schools.com/xml/tempconvert.asmx). The purpose is to ensure accurate temperature conversion from Celsius to Fahrenheit.

**2. API Overview**

* **WSDL:** https://www.w3schools.com/xml/tempconvert.asmx?WSDL
* **Endpoint:** https://www.w3schools.com/xml/tempconvert.asmx
* **SOAP Action:** https://www.w3schools.com/xml/CelsiusToFahrenheit
* **Method:** POST
* **Request/Response Format:** SOAP 1.2 XML

**3. Test Objectives**

* Validate correct Fahrenheit output for valid Celsius inputs.
* Ensure the API handles edge cases, such as very high/low or decimal values.
* Confirm behavior for invalid input (e.g., text instead of number).
* Check SOAP compliance and XML structure correctness.

**4. Test Environment**

* **Tools:** SoapUI, Postman (with SOAP plugin), ReadyAPI, or custom scripts using Python (Zeep), Java (JAX-WS), etc.
* **Auth:** None required
* **Dependencies:** Internet access to public API

**5. Test Scope**

**✅ In Scope:**

* Functional testing of CelsiusToFahrenheit operation
* Boundary testing for numeric inputs
* SOAP structure validation

**❌ Out of Scope:**

* FahrenheitToCelsius method (not in current test scope)
* Load or performance testing

**6. Assumptions**

* API is publicly accessible and stable.
* W3Schools backend will respond within reasonable limits to test data.
* The conversion formula is:

F=(C×9/5)+32F = (C × 9/5) + 32F=(C×9/5)+32

**7. Sample Test Case Detail – TC001**

| **Field** | **Value** |
| --- | --- |
| **Test Case ID** | TC001 |
| **Title** | Convert positive integer (20°C) |
| **Pre-Conditions** | Internet access, SOAP client configured |
| **Input Request** |  |

<?xml version="1.0" encoding="utf-8"?>

<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">

<soap12:Body>

<CelsiusToFahrenheit xmlns="https://www.w3schools.com/xml/">

<Celsius>20</Celsius>

</CelsiusToFahrenheit>

</soap12:Body>

</soap12:Envelope>

| **Expected Response** |

CelsiusToFahrenheitResult>68</CelsiusToFahrenheitResult>

| **Assertions** |

* HTTP Status Code = 200
* XPath: /soap:Envelope/soap:Body/CelsiusToFahrenheitResponse/CelsiusToFahrenheitResult = 68  
  | **Post-Conditions** | None (stateless)

**9. Test Data Table**

| **Celsius** | **Expected Fahrenheit** |
| --- | --- |
| 20 | 68 |
| 0 | 32 |
| -40 | -40 |
| 1000 | 1832 |
| 36.6 | 97.88 |
| 0.123456 | 32.2222208 |

**10. Execution Strategy**

* **Manual Run:** Use SoapUI or Postman to execute each request and verify response manually.
* **Automation Option:** Script using Python (zeep), Java (SoapUI test runner), or Postman Collection runner.

**11. Risks and Mitigations**

| **Risk** | **Mitigation** |
| --- | --- |
| API may be temporarily unavailable | Implement retry logic and test during working hours |
| API response may change over time | Snapshot and verify expected values regularly |
| Floating-point precision differences | Use rounding to 2 decimal places during assertion |

**12. Deliverables**

* Test Plan document (this)
* Test Case Excel or CSV
* Postman/SOAP UI collection (if requested)
* Optional automation script