

Test Case 1 : Valid XML Data

- **Objective :** Ensure the XML data conforms to the XSD schema and can be successfully transformed into HTML.
- **XML Data :**
 - Book Title : Include valid book entries with proper ISBN, titles, authors, publishers, and years.
 - Expected Result : XML should validate against the XSD without errors, and the HTML output should correctly display the data.

Test Case 2: Invalid Year Format

- **Objective:** Check that the XSD schema correctly identifies invalid data in the year element.
- **XML Data:**
 - Modify the year element to include non-numeric characters, e.g., 2065 instead of 2020.
 - **Expected Result:** The XML should fail validation against the XSD schema, generating an error.

Test Case 3: Missing Required Attribute

- **Objective:** Verify that the schema validation fails when a required attribute (e.g., id for book) is missing.
- **XML Data:**
 - Remove the id attribute from one of the book elements.
 - **Expected Result:** The XML should fail validation, indicating that a required attribute is missing.

Test Case 4: Valid XML with Extra Element

- **Objective:** Test if the XSL transformation handles extra elements that are not defined in the XSD schema.
- **XML Data:**
 - Add a new element, e.g., <genre>, to one of the book elements.
 - **Expected Result:** The XML should still validate, but the extra element should not be included in the HTML output unless explicitly handled by the XSL.

Test Case 5: Valid XML but with Wrong Order

- **Objective:** Ensure that the XML data validates even if the order of elements is different from the XSD schema.
- **XML Data:**
 - Change the order of elements within a book entry.
 - **Expected Result:** The XML should validate, as the XSD schema does not enforce element order (assuming the schema allows unordered elements).

Overview

This document explains the transformation of XML data representing a library catalogue into an HTML format using XSLT, and the validation of this XML data against an XSD schema.

Purpose

The main goal is to provide a user-friendly display of the library's book catalogue on a web page and to ensure that the data meets predefined structural and content rules.

XML File (library.xml)

The XML file contains a list of books, each represented with attributes and elements such as id, isbn, title, author, publisher, and year.

XSL Stylesheet (transform.xsl)

The XSL stylesheet transforms the XML data into HTML. The transformation process is defined in the XSL file and includes the creation of a table where each book's details are displayed.

XSD Schema (librdef.xsd)

The XSD schema enforces rules and constraints on the XML data. For example, it ensures that:

- The year element contains only numeric values.
- The id attribute is mandatory for each book element.

- Elements such as isbn, title, author, publisher, and year must be present within each book entry.

Transformation Process

1. **Setup:** Ensure the XML, XSL, and XSD files are correctly linked and accessible.
2. **Transformation:** Use an XSLT processor to apply the XSL stylesheet to the XML data.
3. **Validation:** Validate the XML data against the XSD schema using tools like lxml in Python.
4. **Output:** The result is an HTML file displaying the library catalogue in a table format.

Validation Process

The XML file is validated against the XSD schema to ensure data integrity. If the XML does not conform to the rules defined in the schema, validation will fail, indicating where the issues lie.

Tools Used

- **XSLT Processor:** For transforming XML to HTML (e.g., xsltproc, lxml in Python).
- **XML Schema Validator:** To check the XML data against the XSD schema (e.g., lxml in Python).

Transformation and Validation Process

The XML data representing the library's book catalogue was successfully transformed into an HTML format using XSLT. The XSL stylesheet effectively created an HTML table displaying book details. Validation against the XSD schema ensured the data's integrity, with specific rules such as the requirement for a valid year and mandatory id attributes for each book.

Issues Encountered

- **Invalid Year Format:** In Test Case 2, the XML validation failed when the year element contained non-numeric characters, as expected.
- **Missing Required Attribute:** In Test Case 3, validation failed when the id attribute was removed from a book element, indicating the importance of this attribute.
- **Handling Extra Elements:** In Test Case 4, extra elements not defined in the XSD were ignored by the XSL transformation but did not cause validation errors.
- **Order of Elements:** The order of elements within book entries did not affect validation, as the XSD schema was designed to allow flexibility in element order.

Lessons Learned

- The importance of a well-defined XSD schema for data validation is crucial in maintaining data integrity.
- The XSL transformation process should be adaptable to handle potential variations in the XML structure, such as extra elements or different element orders.
- Clear documentation of the entire process is essential for troubleshooting and future maintenance.

Next Steps

- **Enhancements:** Consider expanding the XSL stylesheet to handle additional elements (e.g., genre) if required in the future.
- **Robust Testing:** Continue testing with more complex scenarios, such as nested elements or more stringent data constraints.
- **Documentation:** Maintain and update the documentation as the XML structure evolves or if new validation rules are introduced.