

XML Transformation and Validation Documentation

PRANAB RAI 2447137

August 2024

1 Introduction

In this documentation how XML data conversion with XSLT (Extensible Stylesheet Language Transformations) and validation by XSD (XML Schema Definition) has been explained. The document is intended to show some experience on working with XML technologies like transforming and validating data and also including any kind of issues faced during the process.

2 Purpose of Files

- `course.xml`: This file contains the XML data representing courses, users, quizzes, certificates, and feedback information.
- `course.xsd`: The XSD file defines the schema for the XML data. It includes constraints, data types, and structure that the XML file must stick to for validation.
- `transform.xsl`: This XSLT file is used to transform the XML data into an HTML format for display in a web browser.
- `course.dtd`: The Document Type Definition (DTD) file defines the structure of the XML data, including the elements, attributes, and entities.

3 Transformation Process

The transformation process involves applying the *transform.xsl* stylesheet to the *course.xml* file to convert the XML data into an HTML format. This allows for easy visualization of the data in a tabular format in a web browser and allows all the users to access and understand data in a much better way.

The *transform.xsl* file is loaded along with the *course.xml* file. The XSLT processor reads the XML data and applies the rules defined in the XSL stylesheet. The transformed output is an HTML document displaying the course, user, quiz, certificate, and feedback data in tables.

4 Validation Process

The validation process ensures that the XML data sticks to the rules defined in the *course.xsd* schema. The *course.xsd* file defines the structure, data types, and constraints

for the XML data. The XML data is validated using this schema to check for any discrepancies or errors. If the XML data does not stick to the schema, validation errors are reported.

4.1 Testing Scenarios

Various scenarios were tested to validate the XML data and ensure proper transformation:

Valid Data : The XML data was first tested with correct data that conforms to the schema. The validation was successful, and the data was transformed into HTML as expected.

Invalid Data: The XML file was deliberately modified to introduce errors that violate the schema rules.

For example:

- An invalid *user-id* format.
- A *date* element with an incorrect date format like- 20/2008/12.
- A *payment* element with an invalid ‘paid’ attribute value like Net Banking.

These errors were detected during the validation process, and appropriate validation errors were reported.

4.2 Errors Encountered

During the validation process, several errors were encountered when the XML data did not conform to the schema some are listed below:

- Invalid *user-id* Format: An incorrect format in the *user-id* field resulted in a validation error. The schema requires a specific pattern for *user-id*, and any deviation from this pattern triggered an error.
- Invalid Date Format: The *date* element was tested with an invalid date format (e.g., ‘YYYY/MM/DD’ instead of ‘YYYY MM DD’). The validation process detected this error and reported it.
- Invalid ‘paid’ Attribute: The *paid* attribute in the *payment* element was tested with an invalid value (e.g., ‘Yes’ instead of ‘Credit-Card’ or ‘EMI’). This caused a validation error as the attribute did not match the allowed values defined in the schema.

5 Conclusion

The XML transformation and validation process was successful in teaching the various dos and don'ts while working on XML, XSLT, and XSD technologies. The transformation process allowed the XML data to be visualized in an HTML format, while the validation process ensured that the XML data conformed to the defined schema. Any errors or issues encountered during validation were detected and reported, highlighting the importance of proper schema definition and validation.