SWT32051-Partical for Servise oriented web application

Day(2025.09.15)

LabSheet Url : <https://drive.google.com/drive/folders/11Vz9K6chlePcFU2_mTcVuQng9cAHBz1t>

**🔹 Understanding XML Parser**

An **XML Parser** is a software component that reads XML documents and provides access to their content and structure.

**Types of XML Parsers:**

1. **DOM (Document Object Model) Parser**
   * Loads the entire XML document into memory as a tree structure.
   * Allows for traversal and modification.
   * Suitable for small to medium XML documents.
2. **SAX (Simple API for XML)**
   * Event-driven.
   * Parses XML document sequentially and triggers events (e.g., startElement, endElement).
   * Faster and more memory-efficient.
   * Read-only and forward-only.
3. **StAX (Streaming API for XML)**
   * Pull-based parser.
   * You write code to pull events when needed.
   * Combines flexibility of DOM and efficiency of SAX.

**🔹 Getting Knowledge of XML DOM (Document Object Model)**

The **XML DOM** is a programming interface for XML documents. It represents the document as a **tree structure**, where each element is an object.

**DOM Tree Structure:**

<book>

<title>XML Guide</title>

<author>John Smith</author>

</book>

Tree:

Document

└── <book>

├── <title>XML Guide</title>

└── <author>John Smith</author>

**Core Concepts:**

* **Document** – Root of the tree
* **Element** – Tags like <book>, <title>
* **Text** – Content inside elements
* **Attributes** – e.g., <book id="101">

**🔹 Syntax, Rules, and Structure of XML DOM**

**XML Syntax Rules:**

1. XML must have a single **root element**.
2. Tags must be properly **nested** and **closed**.
3. XML is **case-sensitive**.
4. Attribute values must be **quoted**.
5. Special characters like <, >, & must be escaped.

**XML DOM Structure:**

* **Elements**: Nodes that can contain attributes, other elements, or text.
* **Attributes**: Metadata inside tags.
* **Text Nodes**: Actual data between the tags.
* **Comments**: <!-- This is a comment -->

**🔹 Implementation of XML DOM (Example in Python)**

Using **Python** and xml.dom.minidom:

**Example XML (books.xml):**

<library>

<book>

<title>XML Basics</title>

<author>Jane Doe</author>

</book>

</library>

**Python Code to Parse:**

from xml.dom.minidom import parse

# Load and parse the XML file

dom = parse("books.xml")

# Get root element

library = dom.documentElement

# Get all book elements

books = library.getElementsByTagName("book")

for book in books:

title = book.getElementsByTagName("title")[0].firstChild.nodeValue

author = book.getElementsByTagName("author")[0].firstChild.nodeValue

print(f"Title: {title}, Author: {author}")

**Output:**

Title: XML Basics, Author: Jane Doe

**✅ Summary**

|  |  |
| --- | --- |
| **Concept** | **Description** |
| **XML Parser** | Software to read and process XML |
| **DOM Parser** | Loads full XML as a tree structure |
| **XML DOM** | API to navigate and manipulate XML |
| **Syntax Rules** | Case-sensitive, well-formed, nested tags |
| **Implementation** | Available in languages like Python, Java, JavaScript, etc. |

XML Parser

* The XML DOM (Document Object Model) defines the properties and methods for
* accessing and editing XML.
* However, to do the tasks, the XML file must be loaded to the XML DOM object.
* XML Parser is used to load the respective XML file to the XML DOM object.
* All major browsers have a built-in XML parser.

**🔸 Code:**

<html>

<body>

<p id = "txtNotice"></p>

<script>

var text, parser, xmlDoc;

text = "<bookstore><book>" +

"<title>Everyday Italian</title>" +

"<author>Giada De Laurentiis</author>" +

"<year>2005</year>" +

"</book></bookstore>";

parser = new DOMParser();

xmlDoc = parser.parseFromString(text, "text/xml");

</script>

</body>

</html>

**✅ Explanation (line by line):**

**<html> <body>**

* This starts the HTML document and the body section where content appears.

**<p id = "txtNotice"></p>**

* This is an empty paragraph (<p>) with an ID.
* You can use JavaScript to show text inside this later using the ID txtNotice.

**<script>**

* This starts the JavaScript block. The code inside runs when the page loads.

**var text, parser, xmlDoc;**

* This line declares **three variables**:
  + text: to hold the XML data as a string.
  + parser: to hold the XML parser object.
  + xmlDoc: to hold the parsed XML document (as a DOM object).

**text = "<bookstore><book>" + ... + "</book></bookstore>";**

* This creates a **string** that looks like an XML file.
* It describes a bookstore with one book.
* The book has a:
  + <title>: Everyday Italian
  + <author>: Giada De Laurentiis
  + <year>: 2005

**parser = new DOMParser();**

* This creates a **new XML parser** object using JavaScript’s built-in DOMParser class.
* This object can read (parse) XML strings.

**xmlDoc = parser.parseFromString(text, "text/xml");**

* This line **parses** the XML string stored in text.
* It converts the text into an **XML DOM object**.
* "text/xml" tells the parser that the input is XML format.
* The result is saved in xmlDoc.

**</script> </body> </html>**

* Closes the script, body, and HTML tags.

**🧠 What does this code do?**

* It creates XML data inside JavaScript (in a string).
* Then it uses DOMParser to convert the string into a real XML DOM object.
* Now, you can **read or access the XML data** using JavaScript (e.g., get the title or author).

✅ This code **only loads the XML into memory** — it doesn’t display anything on the page yet.  
If you want to display the title or other info, you'd need to add extra code using innerHTML or similar.

XML DOM (Document Object Model)

* DOM stands for Document Object Model.
* The XML DOM defines a standard way for accessing and manipulating XML
* documents.
* It presents an XML document as a tree-structure.
* Saying conveniently, XML DOM is a standard for how to get, change, add or delete
* XML elements.
* The XML DOM views an XML document as a tree-structure. The tree structure is called
* a node-tree.
* The tree content can be modified or deleted, and new elements can be created.
* All XML elements can be accessed through the XML DOM.
* XML DOM is a standard programming interface for XML.
* XML DOM is a platform and language independent.

**Summary**

* **DOM** stands for **Document Object Model**.
* It provides a **standard way to access and manipulate** XML documents.
* XML is represented as a **tree structure** called a **node-tree**.
* You can **get, change, add, or delete** XML elements using the DOM.
* Every part of the XML document (elements, attributes, text) is a **node** in the tree.
* The DOM allows **modifying, deleting, or creating** elements.
* It is a **standard programming interface** for working with XML.
* XML DOM is **platform and language independent**, meaning it can be used in any programming environment.

1. Get the value of an XML elements

The following code retrieves the text value of the first <title> element in an XML element.

xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;

2. Loading an XML String

The following program loads a text string into an XML DOM object and extracts the info

from it with JavaScript programming.

<html>

<body>

<p id = "txtNotice"></p>

<script>

var text, parser, xmlDoc;

text = "<bookstore><book>" +

"<title>Everyday Italian</title>" +

"<author>Giada De Laurentiis</author>" +

"<year>2005</year>" +

"</book></bookstore>";

parser = new DOMParser();

xmlDoc = parser.parseFromString(text, "text/xml");

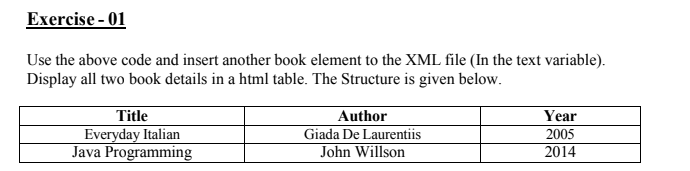
document.getElementById("txtNotice").innerHTML =

xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;

</script>

</body>

</html>



Code

01.

<html>

<body>

<div>

<table border="2">

<tr>

<th>Title</th>

<th>Author</th>

<th>Year</th>

</tr>

<tr>

<td id = "book\_title1"></td>

<td id = "book\_author1"></td>

<td id = "book\_year1"></td>

</tr>

<tr>

<td id = "book\_title2"></td>

<td id = "book\_author2"></td>

<td id = "book\_year2"></td>

</tr>

</table>

</div>

<script>

var text, parser, xmlDoc;

text = "<bookstore>"+

"<book>" +

"<title>Everyday Italian</title>" +

"<author>Giada De Laurentiis</author>" +

"<year>2005</year>" +

"</book>"+

"<book>" +

"<title>Java Programming</title>" +

"<author>John willson</author>" +

"<year>2014</year>" +

"</book>"+

"</bookstore>";

parser = new DOMParser();

xmlDoc = parser.parseFromString(text, "text/xml");

document.getElementById("book\_title1").innerHTML =

xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;

document.getElementById("book\_author1").innerHTML =

xmlDoc.getElementsByTagName("author")[0].childNodes[0].nodeValue;

document.getElementById("book\_year1").innerHTML =

xmlDoc.getElementsByTagName("year")[0].childNodes[0].nodeValue;

document.getElementById("book\_title2").innerHTML =

xmlDoc.getElementsByTagName("title")[1].childNodes[0].nodeValue;

document.getElementById("book\_author2").innerHTML =

xmlDoc.getElementsByTagName("author")[1].childNodes[0].nodeValue;

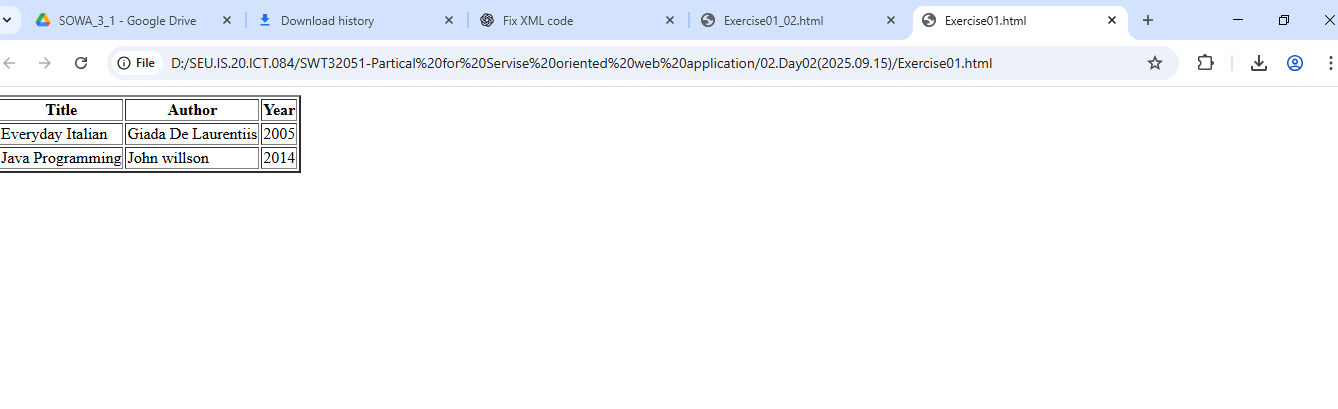
document.getElementById("book\_year2").innerHTML =

xmlDoc.getElementsByTagName("year")[1].childNodes[0].nodeValue;

</script>

</body>

</html>



02.

<html>

<body>

<div>

<table border="2" id="book\_table">

<tr>

<th>Title</th>

<th>Author</th>

<th>Year</th>

</tr>

<!-- Rows will be added here -->

</table>

</div>

<script>

var text, parser, xmlDoc;

text = "<bookstore>"+

"<book>" +

"<title>Everyday Italian</title>" +

"<author>Giada De Laurentiis</author>" +

"<year>2005</year>" +

"</book>"+

"<book>" +

"<title>Java Programming</title>" +

"<author>John willson</author>" +

"<year>2014</year>" +

"</book>"+

"</bookstore>";

parser = new DOMParser();

xmlDoc = parser.parseFromString(text, "text/xml");

var books = xmlDoc.getElementsByTagName("book");

var table = document.getElementById("book\_table");

for (var i = 0; i < books.length; i++) {

var title = books[i].getElementsByTagName("title")[0].textContent;

var author = books[i].getElementsByTagName("author")[0].textContent;

var year = books[i].getElementsByTagName("year")[0].textContent;

var row = table.insertRow(-1);

var cell1 = row.insertCell(0);

var cell2 = row.insertCell(1);

var cell3 = row.insertCell(2);

cell1.innerHTML = title;

cell2.innerHTML = author;

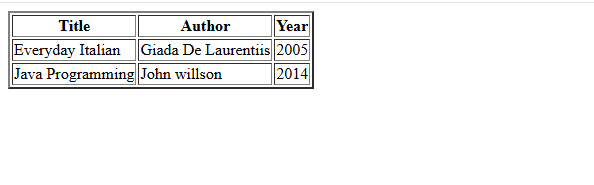
cell3.innerHTML = year;

}

</script>

</body>

</html>



03.

**increase each price by 10%** and **display all book details** (title, author, year, category, updated price), you just need to:

**✅ Steps:**

1. Parse the XML.
2. Loop through each <book>.
3. Extract all needed fields (category, title, author, year, price).
4. Increase the price by 10%.
5. Format the result in a table row.

Code:

<html>

<body>

<table border="2">

<tr>

<th>Title</th>

<th>Author</th>

<th>Year</th>

<th>Category</th>

<th>Updated Price (+10%)</th>

</tr>

<tbody id="update\_price"></tbody>

</table>

<script>

var text, parser, xmlDoc;

text = "<bookstore> "+

"<book category='Entertainment'> "+

"<title lang='En'>Harry Potter</title> "+

"<Author>JK. Rowling</Author> "+

"<Year>2009</Year> "+

"<Price>$45.99</Price>"+

"</book> "+

"<book category='Fiction'> "+

"<title lang='En'>Avengers</title> "+

"<Author>Warner Bros</Author> "+

"<Year>2010</Year> "+

"<Price>$32.89</Price> "+

"</book>"+

"<book category='Mysery'> "+

"<title lang='Sp'>The Great Gatsby</title> "+

"<Author>F Scott Fitzgerald</Author> "+

"<Year>1920</Year> "+

"<Price>$1.99</Price> "+

"</book> "+

"<book category='Historical'> "+

"<title lang='Fr'>Things Fall Apart</title> "+

"<Author>Chinua Achebe</Author>"+

"<Year>1958</Year> "+

"<Price>$12.99</Price> "+

"</book> "+

"<book category='Tragedy'>"+

"<title lang='Es'>Hamlet</title>"+

"<Author>William Shakespeare</Author> "+

"<Year>1601</Year> "+

"<Price>$24.99</Price>"+

"</book> "+

"</bookstore>";

parser = new DOMParser();

xmlDoc = parser.parseFromString(text, "text/xml");

var books = xmlDoc.getElementsByTagName("book");

var tableBody = document.getElementById("update\_price");

var rows = "";

for (var i = 0; i < books.length; i++) {

var title = books[i].getElementsByTagName("title")[0].childNodes[0].nodeValue;

var author = books[i].getElementsByTagName("Author")[0].childNodes[0].nodeValue;

var year = books[i].getElementsByTagName("Year")[0].childNodes[0].nodeValue;

var category = books[i].getAttribute("category");

var priceStr = books[i].getElementsByTagName("Price")[0].childNodes[0].nodeValue;

// Remove $ sign and convert to number

var price = parseFloat(priceStr.replace('$', ''));

var updatedPrice = (price \* 1.10).toFixed(2); // +10%

rows += "<tr>" +

"<td>" + title + "</td>" +

"<td>" + author + "</td>" +

"<td>" + year + "</td>" +

"<td>" + category + "</td>" +

"<td>$" + updatedPrice + "</td>" +

"</tr>";

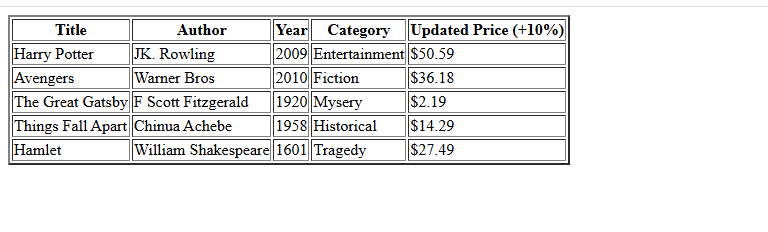
}

tableBody.innerHTML = rows;

</script>

</body>

</html>



<html>

<body>

<div id="content"></div>

<script>

var text, parser, xmlDoc;

text ="<bookstore> "+

"<book category='Entertainment'> "+

"<title lang='En'>Harry Potter</title> "+

"<Author>JK. Rowling</Author> "+

"<Year>2009</Year> "+

"<Price>$45.99</Price>"+

"</book> "+

"<book category='Fiction'> "+

"<title lang='En'>Avengers</title> "+

"<Author>Warner Bros</Author> "+

"<Year>2010</Year> "+

"<Price>$32.89</Price> "+

"</book>"+

"<book category='Mysery'> "+

"<title lang='Sp'>The Great Gatsby</title> "+

"<Author>F Scott Fitzgerald</Author> "+

"<Year>1920</Year> "+

"<Price>$1.99</Price> "+

"</book> "+

"<book category='Historical'> "+

"<title lang='Fr'>Things Fall Apart</title> "+

"<Author>Chinua Achebe</Author>"+

"<Year>1958</Year> "+

"<Price>$12.99</Price> "+

"</book> "+

"<book category='Tragedy'>"+

"<title lang='Es'>Hamlet</title>"+

"<Author>William Shakespeare</Author> "+

"<Year>1601</Year> "+

"<Price>$24.99</Price>"+

"</book> "+

"</bookstore>";

parser = new DOMParser();

xmlDoc = parser.parseFromString(text, "text/xml");

var x = xmlDoc.getElementsByTagName("book");

for (let i = 0; i < x.length; i++){

// Fix: Use getElementsByTagName to access Price tag

let priceTag = x[i].getElementsByTagName("Price")[0];

let oldPrice = parseFloat(priceTag.textContent.replace("$",""));

let newPrice = (oldPrice \* 1.10).toFixed(2); // Fix: remove extra closing parenthesis

priceTag.textContent = "$" + newPrice;

}

let result = "Updated price:\n";

for(let j = 0; j < x.length; j++){

let priceTag = x[j].getElementsByTagName("Price")[0];

result += (j + 1) + " . " + priceTag.textContent + "\n";

}

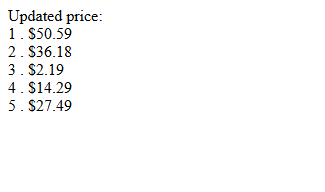
// Fix: Corrected getElementById (was getElementsById)

document.getElementById("content").innerText = result;

</script>

</body>

</html>



**04.display all book data** in the table **but exclude any books published before the year 2000**. That's a simple filter — we just need to **skip books with Year < 2000** during the loop.

Here’s the **modified version** of your code with that change added (and nothing else changed unless necessary):

Code:

<html>

<body>

<table border="2">

<tr>

<th>Title</th>

<th>Author</th>

<th>Year</th>

<th>Category</th>

<th>Updated Price (+10%)</th>

</tr>

<tbody id="update\_price"></tbody>

</table>

<script>

var text, parser, xmlDoc;

text = "<bookstore> "+

"<book category='Entertainment'> "+

"<title lang='En'>Harry Potter</title> "+

"<Author>JK. Rowling</Author> "+

"<Year>2009</Year> "+

"<Price>$45.99</Price>"+

"</book> "+

"<book category='Fiction'> "+

"<title lang='En'>Avengers</title> "+

"<Author>Warner Bros</Author> "+

"<Year>2010</Year> "+

"<Price>$32.89</Price> "+

"</book>"+

"<book category='Mysery'> "+

"<title lang='Sp'>The Great Gatsby</title> "+

"<Author>F Scott Fitzgerald</Author> "+

"<Year>1920</Year> "+

"<Price>$1.99</Price> "+

"</book> "+

"<book category='Historical'> "+

"<title lang='Fr'>Things Fall Apart</title> "+

"<Author>Chinua Achebe</Author>"+

"<Year>1958</Year> "+

"<Price>$12.99</Price> "+

"</book> "+

"<book category='Tragedy'>"+

"<title lang='Es'>Hamlet</title>"+

"<Author>William Shakespeare</Author> "+

"<Year>1601</Year> "+

"<Price>$24.99</Price>"+

"</book> "+

"</bookstore>";

parser = new DOMParser();

xmlDoc = parser.parseFromString(text, "text/xml");

var books = xmlDoc.getElementsByTagName("book");

var tableBody = document.getElementById("update\_price");

var rows = "";

for (var i = 0; i < books.length; i++) {

var year = parseInt(books[i].getElementsByTagName("Year")[0].childNodes[0].nodeValue);

if (year < 2000) {

continue;

}

var title = books[i].getElementsByTagName("title")[0].childNodes[0].nodeValue;

var author = books[i].getElementsByTagName("Author")[0].childNodes[0].nodeValue;

var category = books[i].getAttribute("category");

var priceStr = books[i].getElementsByTagName("Price")[0].childNodes[0].nodeValue;

var price = parseFloat(priceStr.replace('$', ''));

var updatedPrice = (price \* 1.10).toFixed(2);

rows += "<tr>" +

"<td>" + title + "</td>" +

"<td>" + author + "</td>" +

"<td>" + year + "</td>" +

"<td>" + category + "</td>" +

"<td>$" + updatedPrice + "</td>" +

"</tr>";

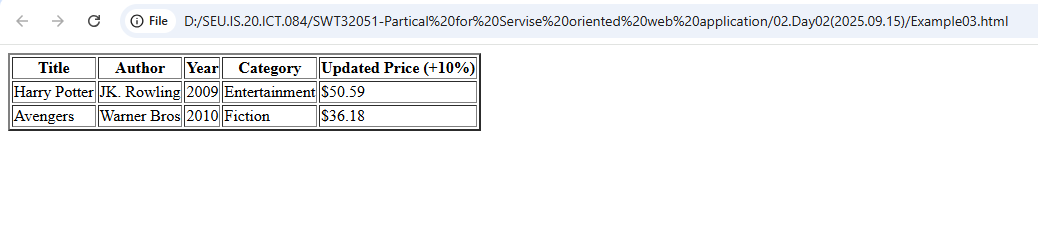
}

tableBody.innerHTML = rows;

</script>

</body>

</html>



ii.

<html>

<body>

<table border="2">

<tr>

<th>Title</th>

<th>Author</th>

<th>Year</th>

<th>Category</th>

<th>Original Price</th>

</tr>

<tbody id="update\_price"></tbody>

</table>

<script>

var text, parser, xmlDoc;

text = "<bookstore> "+

"<book category='Entertainment'> "+

"<title lang='En'>Harry Potter</title> "+

"<Author>JK. Rowling</Author> "+

"<Year>2009</Year> "+

"<Price>$45.99</Price>"+

"</book> "+

"<book category='Fiction'> "+

"<title lang='En'>Avengers</title> "+

"<Author>Warner Bros</Author> "+

"<Year>2010</Year> "+

"<Price>$32.89</Price> "+

"</book>"+

"<book category='Mysery'> "+

"<title lang='Sp'>The Great Gatsby</title> "+

"<Author>F Scott Fitzgerald</Author> "+

"<Year>1920</Year> "+

"<Price>$1.99</Price> "+

"</book> "+

"<book category='Historical'> "+

"<title lang='Fr'>Things Fall Apart</title> "+

"<Author>Chinua Achebe</Author>"+

"<Year>1958</Year> "+

"<Price>$12.99</Price> "+

"</book> "+

"<book category='Tragedy'>"+

"<title lang='Es'>Hamlet</title>"+

"<Author>William Shakespeare</Author> "+

"<Year>1601</Year> "+

"<Price>$24.99</Price>"+

"</book> "+

"</bookstore>";

parser = new DOMParser();

xmlDoc = parser.parseFromString(text, "text/xml");

var books = xmlDoc.getElementsByTagName("book");

var tableBody = document.getElementById("update\_price");

var rows = "";

for (var i = 0; i < books.length; i++) {

var year = parseInt(books[i].getElementsByTagName("Year")[0].childNodes[0].nodeValue);

if (year < 2000) {

continue;

}

var title = books[i].getElementsByTagName("title")[0].childNodes[0].nodeValue;

var author = books[i].getElementsByTagName("Author")[0].childNodes[0].nodeValue;

var category = books[i].getAttribute("category");

var priceStr = books[i].getElementsByTagName("Price")[0].childNodes[0].nodeValue;

rows += "<tr>" +

"<td>" + title + "</td>" +

"<td>" + author + "</td>" +

"<td>" + year + "</td>" +

"<td>" + category + "</td>" +

"<td>" + priceStr + "</td>" +

"</tr>";

}

tableBody.innerHTML = rows;

</script>

</body>

</html>

