Assignment of Web Engineering

Submitted to:

Professor Dr. Ather Ashraf

Submitted by:

Saad Ishtiaq

Roll no: BSEF18A025

Punjab University College of Information Technology,

PUCIT (Old Campus),

LAHORE

Introduction:

You know , reading from memory and reading from disk , which one is faster. I think, everyone says memory will be faster. And you know Tree data structure is faster. So, DOM is inspired from inMemory loading and Tree datastructure. Both the parser work in different way internally ,but intent of both are same. Internal implementation of DOM Vs SAX are different. It means, with same intent philosophy of the implementation are different.

SAX Parser

The SAX or Simple API (Application programming interface) for XML is a parser which is used to parse the XML documents using a sequence of occurrences called "events".

This parser requires a good interaction among the application program and the parser itself since it requires repeated event handling by the parser. Once the parser reads the XML document in a top to bottom fashion, i.e., starting from the beginning of the code and traversing in the left to right fashion towards the bottom; tokens are reported to the application program which in turn send an event handler to be used with the parser. The tokens are identified thereafter and functions in the handler are called.

DOM Parser

The DOM or the Document Object Model can be comparatively easier to understand just because of the familiarity with its structure. It simply converts all the elements of our document into a tree structure. The node of the tree contains data elements. The tree also lets us traverse the structure in an easy manner and perform search and modify operations.

Key Difference of DOM and SAX

- **DOM** stands for Document Object Model while **SAX** stands for Simple API for XML parsing.
- DOM parser load full XML file in memory and creates a tree representation of XML document, while SAX is an event based XML parser and doesn't load whole XML document into memory.
- DOM parser load entire XML file in memory and creates a tree structure of XML document, while SAX is an event based XML parser and doesn't load whole XML document into memory
- If you know you have sufficient amount of memory in your server you can choose DOM as this faster because load entire xml in memory and works as tree structure which is faster to access.
- As a thumb rule, for small and medium sized XML documents, DOM is much faster than SAX because of in memory agnostic.

Here is extended information about the differences among the two.

SAX vs DOM Parser

Point of Difference	SAX	DOM
First published	The SAX parser was introduced particularly after the DOM originated.	October 1, 1998.
Abbreviated for	Simple API for XML	Document object model.
Loading	Loads a part of the document in the memory at one time.	Loads the whole XML document in memory at one go.
A general methodology	Uses the "event handling" method for parsing. Various calls are made to the application program by the parser for acknowledgement of recognition of the tokens created by the parser by reading the document in a top to bottom manner.	Uses the tree methodology. Leads to a tree like structure which contains all the data elements for parsing the document. This structure makes traversing easy.
Parsing	Parses the document only until we want it to.	Parses the whole document together.
Functionality	Provides comparatively less functionalities.	Provides sufficient functionalities with the proper data structures.
Time	The SAX parser is less time consuming.	Consumes more time.
Space	Occupies less memory space since it does not create any internal structure.	Since it creates an internal structure, it occupies more space in memory.
Suitable for	Parsing large documents.	Parsing small programs, data lines or configuration lines.
Nesting	Does not work very well for documents where deep nesting is observed. Also, not suitable for "sorting-like" documents.	Can work considerably well for such documents.
Interface methods	The interface methods are used for accessing the attributes of various elements. Example: intgetLength()- This function returns the number of attributes.	The interface methods are used for traversing the tree and performing modifications. Example: Document.getDocumentElement() – Returns the root element of the document.