XML в Java

XML (Extensible Markup Language) - это SGML-производный язык разметки документов, позволяющий структурировать информацию разного типа, используя для этого произвольный набор инструкций.

XML – это простой способ структурировать данные и обмениваться ими как через интернет, так и в рамках локальных сетей и в целом внутри IT-инфраструктуры.

Спецификация XML описывает XML-документы и частично описывает поведение XML-процессоров (программ, читающих XML-документы и обеспечивающих доступ к их содержимому).

Язык XML называется расширяемым, поскольку он не фиксирует разметку, используемую в документах: разработчик волен создать разметку в соответствии с потребностями к конкретной области, будучи ограниченным лишь синтаксическими правилами языка.

Существует два подхода к построению парсеров XML: SAX-парсеры и DOM-парсеры.

**SAX-парсеры**

SAX (англ. «Simple API for XML») — способ последовательного чтения/записи XML-файлов.

Обычно SAX-парсеры требуют фиксированного количества памяти для своей работы, но не позволяют изменять содержимое документа. Всё, что делает SAX-парсер, это сообщает вызвавшему приложению о встреченных распознанных элементах XML-разметки или о встреченных ошибках. Связь парсера с вызывающим приложением, как правило, осуществляется посредством функций обратного вызова.

Применяются SAX-парсеры либо для быстрого поиска по XML-документам, либо во время построения DOM, либо для чтения XML-потоков большого объема (когда построение DOM требует слишком большого объема памяти).

SAX — событийный парсер XML. Он содержит очень похожую модель обработки событий на такую же модель в Java. К примеру, в апплете можно зарегистрировать обработчик события на клик мышкой, а в SAX можно зарегистрировать обработчик события на начало и конец тегов элементов.

В Java SAX реализован в пакете org.xml.sax.\*.

**DOM-парсеры**

DOM (от англ. Document Object Model — «объектная модель документа») — это не зависящий от платформы и языка программный интерфейс, позволяющий программам и скриптам получить доступ к содержимому HTML, XHTML и XML-документов, а также изменять содержимое, структуру и оформление таких документов.

Аббревиатура DOM - это не парсер, а способ представления уже разобранного XML в виде структуры в памяти.

Парсер, который представляет в виде DOM (древовидной структуры узлов) называется DOM-парсером.

В Java DOM-парсер представлен классами javax.xml.parsers.DocumentBuilderFactory, javax.xml.parsers.DocumentBuilder;

**Сравнение DOM и SAX парсеров**

|  |  |
| --- | --- |
| **SAX** | **DOM** |
| Основан на последовательности событий | Загружает весь файл в память и потом разбирает его |
| Разбирает файл по мере чтения, тег за тегом | Имеет ограничения по памяти, поскольку весь файл загружается перед разбором |
| Не имеет ограничений по памяти, поскольку содержимое файла не храниться в памяти | Позволяет читать, записывать, вставлять и удалять узлы |
| Позволяет только читать XML. Нельзя добавить или удалить узел. | Позволяет проводить поиск в прямом и обратном направлении, просматривать содержимое узлов |
| Читает файл сверху вниз и обратная навигация невозможна | Предоставляет ясную объектную структуру |
| Быстро работает во время выполнения | Более медленный во время выполнения |

Какой парсер лучше использовать в вашем случае - SAX или DOM, можно решить по типу исходных данных. Если исходный документ большой по объему и вложенность элементов невысока - скорее всего SAX, при этом подходе разбор исходных данных происходит несколько быстрее и используется меньше памяти.

Если нужно преобразовать сложную структуру данных - наверное удобнее и проще будет работать с DOM.

1. <https://ru.wikipedia.org/wiki/SAX>
2. <https://ru.wikipedia.org/wiki/Document_Object_Model>
3. <http://khpi-iip.mipk.kharkiv.edu/library/extent/prog/iipXML/xmljava.html>
4. <http://stackoverflow.com/questions/6828703/what-is-the-difference-between-sax-and-dom>
5. <http://www.mkyong.com/java/how-to-read-xml-file-in-java-dom-parser/>
6. <http://www.quizful.net/post/sax-parser-java>