

Выполнил(а) Павличенко С.А., № группы P3115, оценка                       
Фамилия И.О. студента не заполнять

<b>Название статьи/главы книги/видеолекции</b> Как и зачем мы парсим XML		
<b>ФИО автора статьи (или e-mail)</b> @LightKitten	<b>Дата публикации (не старше 2020 года)</b> "26" октября 2023 г.	<b>Размер статьи (от 400 слов)</b> 1485
<b>Прямая полная ссылка на источник или сокращённая ссылка (bit.ly, tr.im и т.п.)</b> <a href="https://habr.com/ru/companies/tensor/articles/769722/">https://habr.com/ru/companies/tensor/articles/769722/</a>		
<b>Теги, ключевые слова или словосочетания</b> XML, Saby, C++, DOM-интерфейс, парсинг		
<b>Перечень фактов, упомянутых в статье (минимум три пункта)</b> <ol style="list-style-type: none"> <li>1. Saby - это экосистема для бизнеса, которая обрабатывает и обменивается электронными документами с различными организациями и государственными органами.</li> <li>2. XML-формат широко используется для обмена данными в Saby, включая отчеты в государственные органы и складские документы.</li> <li>3. Разбор XML-документов может потребовать значительных ресурсов, особенно при обработке больших файлов.</li> <li>4. Изначально в Saby использовались DOM-интерфейсы для разбора XML-документов с библиотекой Xerces-C.</li> <li>5. DOM-интерфейсы позволяют получить полный доступ к XML-дереву, но могут потреблять большое количество памяти, особенно при обработке больших файлов.</li> <li>6. Для обработки больших XML-документов и снижения потребления памяти было принято решение перейти на событийный SAX-парсер, который позволил обрабатывать XML-документы последовательно.</li> <li>7. В статье провели измерения производительности различных XML-парсеров, включая время работы, пиковое потребление оперативной памяти и скорость парсинга.</li> <li>8. Библиотеки, участвующие в сравнении, включают xerces-c, libxml2, pugixml, expat, rapidxml и vtd-xml.</li> <li>9. Функциональное сравнение парсеров также было проведено, включая поддержку пространств имен, processing instruction, кодировок и XPath.</li> </ol>		
<b>Позитивные следствия и/или достоинства описанной в статье технологии (минимум три пункта)</b> <ol style="list-style-type: none"> <li>1. Событийные парсеры позволяют управлять потреблением оперативной памяти, что позволяет обрабатывать большие документы без проблем с памятью.</li> <li>2. Использование событийных SAX-парсеров позволяет обрабатывать XML-документы любого размера, что важно при работе с большими данными.</li> <li>3. Пользователям, работающим с документами в Saby, не нужно заботиться о сложностях парсинга XML, так как это решается на уровне технической реализации.</li> <li>4. Помимо XML, Saby также поддерживает другие форматы файлов, такие как JSON и TXT, что позволяет обрабатывать разнообразные данные.</li> </ol>		
<b>Негативные следствия и/или недостатки описанной в статье технологии (минимум три пункта)</b> <ol style="list-style-type: none"> <li>1. Переход с использования DOM-интерфейсов на событийные SAX-парсеры требует значительных усилий и времени для адаптации существующего кода и механизмов.</li> <li>2. Событийные парсеры ориентированы на последовательное чтение XML и могут быть менее удобными для работы с сложными структурами данных, особенно при необходимости обращения к разным частям документа.</li> <li>3. В статье указано, что некоторые событийные парсеры имеют ограниченную поддержку XPath, что может ограничивать возможности выполнения запросов в сложных документах.</li> </ol>		

