

# JAXP & JAXB

**№ урока:** 2    **Курс:** Основы Java EE

**Средства обучения:** JDK, IntelliJ Idea

## Обзор, цель и назначение урока

Общие понятия JAXP. Разбор SAX, DOM, StAX парсеров. Общие понятия JAXB, XSLT, XPath.

## Изучив материал данного занятия, учащийся сможет:

- Научиться работать с Java и XML
- Научиться парсить с помощью SAX-парсера
- Научиться парсить с помощью DOM-парсера
- Научиться парсить с помощью StAX- парсера
- Научиться парсить XML с помощью JAXB.
- Научиться парсить XML с помощью XPath
- Научиться преобразовывать XML-документ в HTML или в другой XML-документ

## Содержание урока

1. Введения в JAXP
2. Разбор SAX-парсера
3. Разбор DOM-парсера
4. Разбор StAX-парсера
5. Введения в JAXB
6. Парсинг с помощью XPath
7. Понятие XSLT
8. Преобразование XML-документа в HTML средствами XSLT.
9. Парсинг с помощью JAXB

## Резюме

- **JAXP** — набор абстрактных API, упрощающих обработку XML данных с помощью программ, написанных на Java. Усиливает стандартные анализаторы SAX (Simple API for XML Parsing) и DOM (Document Object Model). Начиная с версии 1.1 поддерживает стандарт XSLT (XML Stylesheet Language Transformations). JAXP обеспечивает поддержку пространств имен (namespace), то есть имеется возможность работать с DTD. JAXP дает приложению возможность производить лексический разбор и модификацию документов XML независимо от конкретной реализации обработки XML.  
JAXP — библиотека, предоставляющая доступ ко всем методам программной обработки XML-документов. Входит в базовую поставку Java SE. Включает в себя библиотеки из Apache XML Project — Xerces и Xalan. Это стандарт, под который достаточно просто найти уже написанный парсер, распространяемый свободно.  
JAXP 1.1 Reference Implementation предоставляет XSLT-процессор Xalan и синтаксический анализатор Crimson, разработанные совместно Sun и организацией Apache Software Foundation которая предоставляет программное обеспечение с открытым исходным кодом.  
Начиная с самых ранних версий, JAXP-спецификации, так же, как и лежащие в их основе спецификации XML и XML Namespaces, были привязаны к конкретным версиям DOM и SAX для гарантии максимальной переносимости построенных на их основе приложений. За три года, прошедших с момента выпуска последней основной версии JAXP (JAXP 1.1), ни одна из этих спецификаций не оставалась статичной. JAXP1.3 основан на самых последних версиях каждой из этих спецификаций, позволяя им пройти свой путь к J2SE и J2EE.

- **SAX** — 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-потокa 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.

- **Объектная модель документа (DOM)** - спецификация прикладного программного интерфейса для работы со структурированными документами (как правило, документов XML). Определяется эта спецификация консорциумом W3C.

С точки зрения объектно-ориентированного программирования, DOM определяет классы, методы и атрибуты этих методов для анализа структуры документов и работы с представленными документами в виде дерева. Все это предназначено для того, чтобы предоставить возможность компьютерной программе доступа и динамической модификации структуры, содержания и оформления документа.

Вместе с распространением и развитием веб-технологий и браузеров начали появляться разные, часто несовместимые интерфейсы работы с HTML документами в интерпретаторах JavaScript встроенных в веб-браузере. Это побудило World Wide Web Consortium (W3C) согласовать и определить ряд стандартов, которые получили название W3C Document Object Model (W3C DOM). Спецификации W3C не зависящих от платформы или языка программирования. Так как структура документа представляется в виде дерева, полное содержание документа анализируется и сохраняется в памяти компьютера. Поэтому, DOM подходит для применений в программах, которые требуют многократный доступ к элементам документа в произвольном порядке. В случае, если надо только последовательный или однократный доступ к элементам документа, рекомендуется, для ускорения переработки и уменьшения объемов необходимой памяти компьютера, использовать последовательную модель работы со структурированными документами (SAX).

- С момента своего выхода стандарт **Java-API for XML Processing (JAXP)** предоставляет два метода обработки XML - метод объектной модели документа (**DOM**), который для представления XML-документов использует стандартную объектную модель, и метод **SAX (Simple API for XML, SAX)**, который для обработки XML использует предоставляемые приложениями обработчики событий. Поточная альтернатива этим двум подходам была предложена в стандарте **JSR-173: Streaming API for XML (StAX)**. Окончательная редакция стандарта была опубликована в марте 2004 года и стала частью JAXP 1.4 (которая будет включена в разрабатываемую версию Java 6).

Как можно понять из названия стандарта, **StAX** фокусируется на поточной обработке. По сути, именно возможность приложений обрабатывать данные XML в виде потока событий отличает стандарт StAX от других подходов. Идея обработки XML как набора событий не слишком свежая (по сути, она уже представлена в стандарте SAX); однако в StAX программный код может запрашивать такие события одно за другим вместо того, чтобы предоставлять обработчики, которые получали бы события от синтаксического анализатора в определяемом самим анализатором порядке.

На самом деле **StAX** состоит из двух наборов API для обработки XML, которые обеспечивают разные уровни абстракции. API с использованием курсора позволяет приложениям работать с XML как с потоком лексем (или событий); приложение может проверить статус анализатора и получить информацию о последней проанализированной лексеме, а затем перейти к следующей. Это, скорее, низкоуровневый API; он довольно эффективный, но не предоставляет абстракции нижележащей XML-структуры. Высокоуровневый API, использующий итераторы событий, позволяет приложению обрабатывать XML как серию объектов событий, каждый из которых взаимодействует с фрагментом XML-структуры приложения. Все, что требуется от приложения - это определить тип синтаксически разобранный события, отнести его к

соответствующему конкретному типу и использовать соответствующие методы для получения информации, относящейся к событию.

- **JAXB** предлагает быстрый и удобный способ создания двухстороннего преобразования между XML документами и классами языка Java. Задавая XSD, компилятор JAXB создает набор классов Java, содержащий весь код, который необходим для анализа XML документов, основанных на данной схеме. Разработчик, использующий созданные классы, может строить дерево объектов Java, изображающих XML документ, обрабатывать содержимое этого дерева и заново создавать XML документы из него.

- **XPath** (XML Path Language) — язык запросов к элементам XML-документа. Разработан для организации доступа к частям документа XML в файлах трансформации XSLT и является стандартом консорциума W3C. XPath призван реализовать навигацию по DOM в XML. В XPath используется компактный синтаксис, отличный от принятого в XML. В 2007 году завершилась разработка версии 2.0, которая теперь является составной частью языка XQuery 1.0. В декабре 2009 года началась разработка версии 2.1, которая использует XQuery 1.1.

На данный момент самой популярной версией является XPath 1.0. Это связано с отсутствием поддержки XPath 2.0 со стороны открытых библиотек. В частности, речь идёт о LibXML, от которой зависит поддержка языка в браузерах, с одной стороны, и поддержка со стороны серверного интерпретатора, с другой.

XML имеет древовидную структуру. В документе всегда имеется корневой элемент (инструкция `<?xml version="1.0"?>` к дереву отношения не имеет). У элемента дерева всегда существуют потомки и предки, кроме корневого элемента, у которого предков нет, а также тупиковых элементов (листьев дерева), у которых нет потомков. Каждый элемент дерева находится на определённом уровне вложенности (далее — «уровень»). У элементов на одном уровне бывают предыдущие и следующие «братские» элементы, каждый из которых образует со своими потомками отдельный слой внутри уровня.

Это очень похоже на организацию каталогов в файловой системе, и строка XPath в логическом смысле — правила выбора пути к нужным «файлам» — элементам.

Физически же, на каждом шаге такого пути отбираются элементы, соответствующие условиям отбора на этом шаге, и в результате обращения по пути к документу получается множество элементов, удовлетворяющих данному пути.

- **XSLT** (eXtensible Stylesheet Language Transformations) — язык преобразования XML-документов. Спецификация XSLT входит в состав XSL и является рекомендацией W3C.

При применении таблицы стилей XSLT, состоящей из набора шаблонов, к XML-документу (исходное дерево) образуется конечное дерево, которое может быть сериализовано в виде XML-документа, XHTML-документа (только для XSLT 2.0), HTML-документа или простого текстового файла. Правила выбора (и, отчасти, преобразования) данных из исходного дерева пишутся на языке запросов XPath.

XSLT имеет множество различных применений, в основном в области веб-программирования и генерации отчётов. Одной из задач, решаемых языком XSLT, является отделение данных от их представления, как часть общей парадигмы MVC (англ. Model-view-controller). Другой стандартной задачей является преобразование XML-документов из одной XML-схемы в другую. XSLT разработан рабочей группой XSL Консорциума Всемирной паутины.

Версия 1.0 была одобрена в качестве рекомендации 16 ноября 1999 года. После выхода первой версии начались работы над версией 1.1, но в 2001 году они были прекращены, а рабочая группа XSL присоединилась к рабочей группе XQuery для совместной работы над XPath 2.0. Впоследствии XPath 2.0 послужил основой при разработке XSLT версии 2.0.

Версия 2.0 была одобрена в качестве рекомендации 24 января 2007 года. Несмотря на то, что версия 2.0 является последней, версия 1.0 продолжает использоваться очень широко.

В процессе выполнения XSLT-преобразования задействованы:

- один или несколько входных XML-документов;
- одна или несколько таблиц стилей XSLT;
- XSLT-процессор;
- один или несколько выходных документов.

В простейшем случае XSLT-процессор получает на входе два документа — входной XML-документ и таблицу стилей XSLT — и создаёт на их основе выходной документ.

## Закрепление материала

- Что такое JAXP?
- Как работает SAX-парсер?
- Как работает DOM-парсер?
- Как работает StAX-парсер?
- В чем разница между SAX и StAX парсерами?
- Что такое JAXB?
- Что такое XPath?
- Какие существуют, предикаты, wildcards в XPath?
- Что такое XSLT и как он работает?

## Дополнительное задание

### Задание

Создать xml-документ candies.xml, с корневым элементом **<factory>**, который будет в себе содержать несколько **<candy>**, с характеристиками конфеты. Используя StAX-парсер получить коллекцию Candy из xml-файла.

## Самостоятельная деятельность учащегося

### Задание 1

Выучите основные понятия, рассмотренные на уроке.

### Задание 2

Из задания первого урока на стр.449, которое вы выбрали

[https://careers.epam.by/content/dam/epam/by/book\\_epam\\_by/JAVA\\_Methods\\_Programming\\_v2.march2015.pdf](https://careers.epam.by/content/dam/epam/by/book_epam_by/JAVA_Methods_Programming_v2.march2015.pdf)

Сгенерировать класс, соответствующий данному описанию.

Создать приложение для разбора XML-документа и инициализации коллекции объектов информацией из XML-файла. Для разбора использовать SAX, DOM и StAX парсеры. Для сортировки объектов использовать интерфейс Comparator.

### Задание 3

Из задания первого урока на стр.449, которое вы выбрали

[https://careers.epam.by/content/dam/epam/by/book\\_epam\\_by/JAVA\\_Methods\\_Programming\\_v2.march2015.pdf](https://careers.epam.by/content/dam/epam/by/book_epam_by/JAVA_Methods_Programming_v2.march2015.pdf)

С помощью JAXB распарсить XML-документ.

### Задание 4

Из задания первого урока на стр.449, которое вы выбрали

[https://careers.epam.by/content/dam/epam/by/book\\_epam\\_by/JAVA\\_Methods\\_Programming\\_v2.march2015.pdf](https://careers.epam.by/content/dam/epam/by/book_epam_by/JAVA_Methods_Programming_v2.march2015.pdf)

С помощью XPath распарсить XML-документ.

### Задание 5

Из задания первого урока на стр.449, которое вы выбрали

[https://careers.epam.by/content/dam/epam/by/book\\_epam\\_by/JAVA\\_Methods\\_Programming\\_v2.march2015.pdf](https://careers.epam.by/content/dam/epam/by/book_epam_by/JAVA_Methods_Programming_v2.march2015.pdf)

Определить метод, производящий преобразование разработанного XML-документа в документ, указанный в каждом задании.

### Задание 6

Используя поисковые механизмы <https://www.w3schools.com/> найти материал по теме и закрепить полученные знания.

## Рекомендуемые ресурсы

JAXP

<https://www.jcp.org/en/jsr/detail?id=206>

Другие реализации JAXP

[http://jar.fyicenter.com/1497\\_XML\\_List\\_of\\_Java\\_API\\_for\\_XML\\_Processing\\_JAXP\\_Implementati.html](http://jar.fyicenter.com/1497_XML_List_of_Java_API_for_XML_Processing_JAXP_Implementati.html)

JAXB

<https://www.ibm.com/developerworks/ru/library/x-pracdb2/index.html>

XSLT

[https://www.w3schools.com/xml/xml\\_xslt.asp](https://www.w3schools.com/xml/xml_xslt.asp)

XPath

[https://www.w3schools.com/xml/xml\\_xpath.asp](https://www.w3schools.com/xml/xml_xpath.asp)