XML - код, прийнятий в якості рекомендації Консорціуму World Wide Web (W3C), аналогічний мови HTML. Він є менш складним і зручним у застосуванні, належить до підмножини мови стандарту розмітки SGML, використовуваного для створення структури документа. Основним будівельним блоком XML є елемент, який визначається початковими і кінцевими тегами. Всі дані містяться в документі XML у зовнішньому елементі, відомому як кореневої. Імена описують його вміст. А структура описує відносини між елементами в прикладі XML. Він підтримує вкладені або внутрішні елементи з ієрархічною структурою.

**Історія**

XML з'явився як спосіб подолати недоліки двох своїх попередників - SGML і HTML. В кінці 1980-х до появи Інтернету видавці цифрових ЗМІ реалізували різні переваги SGML для динамічного відображення інформації. Мова був надзвичайно потужним та розширюваним інструментом для семантичної розмітки і особливо корисний для каталогізації та індексації даних. SGML і сьогодні можна використовувати для створення нескінченної кількості мов розмітки.   
  
  
Однак SGML залишається досить складним і дорогим, особливо для повсякденного використання в інтернеті. Додавання можливості SGML до текстового процесору подвоює або навіть влаштовує його ціну. Нарешті, комерційні браузери ясно дали зрозуміти, що вони не мають наміру коли-небудь підтримувати SGML. Одним з найпопулярніших додатків SGML став розвиток мови розмітки гіпертексту HTML, створеного Тімом Бернерсом в кінці 1980-х. З моменту свого розвитку HTML став жертвою власної популярності, так він був швидко прийнятий і розширено багатьма способами, що виходять за рамки його первісного бачення.

[](http://hi-news.pp.ua/uploads/posts/2018-12/xml-priklad-funkcyi-ta-mozhlivost-plyusi-mnusi-formatu_341.jpeg)

Він залишається популярним і сьогодні, хоча вважається невідповідним як формат зберігання даних загального призначення на відміну від інших мов. Приклад: XML усуває прогалину, будучи читання як для людини, так і для комп'ютера, і в той же час досить гнучким для підтримки обміну даними, незалежної від платформи і архітектури. У 1998 році W3C схвалив версію XML 1.0 таким чином, офіційно було створено нову мову.

**Структура елементів**

Існує два способи визначити структуру XML-документа (визначення типу даних (DTD) або схеми XML). Документи були DTD введені SGML. Вони відповідають розширеній формі Backus Naur (EBNF).

[](http://hi-news.pp.ua/uploads/posts/2018-12/xml-priklad-funkcyi-ta-mozhlivost-plyusi-mnusi-formatu_512.jpeg)

Документи XML-схеми пишуться з використанням синтаксису коду. Як DTD, так і XML-схема дозволяє задавати правила обмеження. Вони застосовуються до вмісту документів примірника цього ж коду. Вони приймають форму правил для перевірки XML структури. Всі XML-документи мають один кореневий елемент, який містить подэлементы, їх подэлементы і т. д. Це призводить до ієрархічної деревоподібної структурі в них. Завдяки розробці через SGML визначення типів документів більше підходять для додатків, орієнтованих на документи, таких як HTML. В HTML використовується DTD. Хоча він може визначати структуру документа, але не може визначати правила, які мають застосовуватися до даних. Тобто всі дані, що містяться в XML-документі, в DTD обробляються як рядок. Це підходить для мов розмітки. Але не підходить, коли додатком необхідно контролювати містяться в ньому дані.   
Документ XML вважається «правильно сформованим», тобто може бути прочитаний і зрозумілий синтаксичним аналізатором, якщо його формат відповідає специфікації XML і він правильно розмічений. А елементи належним чином вкладені. Код також підтримує можливість визначення атрибутів для елементів і опису характеристик в початковому тегу. Документи можуть бути дуже простими для XML, приклад тексту "Привіт, світ!":

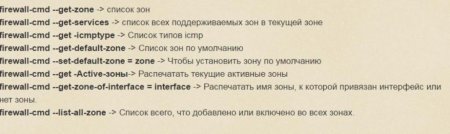
hello world

**Керівництво з безпеки брандмауера**

Функції безпеки і можливості брандмауера XML роблять його цінним і суттєвим доповненням до стратегії веб-служби будь-якої організації. Порівняно з іншими брандмауер XML здатний проводити глибоку перевірку, а також має безліч інших функцій, які роблять його головним конкурентом для захисту даних і запобігання вразливостей і загроз.

Виробники постійно додають нові функції, щоб тримати зловмисників в напрузі і страху, запобігаючи їх шкідливі дії. На жаль, деякі брандмауери на ринку сьогодні все ще знаходяться за межами допустимості, коли мова заходить про захист повідомлень і веб-сервісів. Брандмауери XML захищають більш потужно, ніж традиційні. Приклад XML-файлу, який демонструє роботу мережевого екрану.

<script type="text/jаvascript">   
var blockSettings3 = {blockId:"R-A-271049-6",renderTo:"yandex\_rtb\_R-A-70350-44",async:!0};   
if(document.cookie.indexOf("abmatch=") >= 0) blockSettings3.statId = 70350;   
!function(a,b,c,d,e){a[c]=a[c]||[],a[c].push(function(){Ya.Context.AdvManager.render(blockSettings3)}),e=b.getElementsByTagName("script")[0],d=b.createElement("script"),d.type="text/jаvascript",d.src="//an.yandex.ru/system/context.js",d.async=!0e.parentNode.insertBefore(d,e)}(this,this.document,"yandexContextAsyncCallbacks");

[](http://hi-news.pp.ua/uploads/posts/2018-12/xml-priklad-funkcyi-ta-mozhlivost-plyusi-mnusi-formatu_113.jpeg)

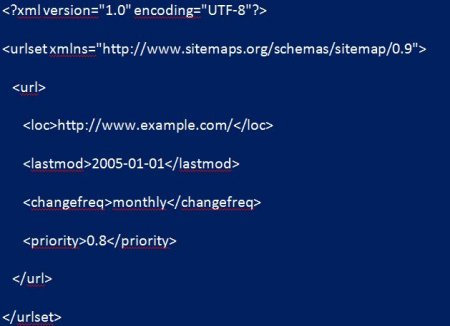
Традиційні брандмауери добре працюють зі звичайним трафіком, але екранування потоків даних вимагає зміни технології, щоб забезпечити захист. Це робить брандмауер XML одним з важливих елементів забезпечення безпеки веб-сервісів.

Підприємства, що розробляють веб-додатків та веб-служб на основі цього коду, все частіше звертаються до мови розмітки тверджень безпеки (SAML) для передачі ідентифікаційних даних та інформації про авторизації, тому їм необхідно захищати себе від атак протоколів XML і SAML на рівні самого додатка. Брандмауер XML може бути адекватним засобом безпеки при захисті багаторівневих систем.

**Управління в SOA**

Багато установки SOA стикаються з проблемами продуктивності, оскільки їм не вистачає належного управління даними. Незважаючи на весь галас і модні слова, які в даний час оточують SOA і прагнуть інтегрувати сервіс-орієнтовану архітектуру в свою ІТ-інфраструктуру, розробники все ще не враховують проблеми, пов'язані з інтеграцією даних та управлінням у їх проектах.

Приклад sitemap XML і нюанси використання демонструє можливості мови.

[](http://hi-news.pp.ua/uploads/posts/2018-12/xml-priklad-funkcyi-ta-mozhlivost-plyusi-mnusi-formatu_384.jpeg)

Суть полягає в тому, щоб розпізнати цінність даних організації, де б вони не знаходилися (під парасолькою SOA чи за його межами) і знайти методи, що дозволяють їм збирати і передавати інформацію між виробниками і споживачами з мінімальною складністю. Приклад XML-файлу SOA для збереження інформації представлений нижче.

[](http://hi-news.pp.ua/uploads/posts/2018-12/xml-priklad-funkcyi-ta-mozhlivost-plyusi-mnusi-formatu_855.jpeg)

Формуючи метадані з використанням XML, а потім, створюючи додатки XSLT для передачі їх компоненти SOA і з них, розробники отримують численні переваги:

1. Вони створюють засоби для захоплення ключових елементів даних, взаємодій і семантики. Вони полегшують переміщення їх між компонентами SOA або між собою. Також документують основні поняття та припущення щодо даних, які вони використовують і необхідних метаданих.
2. Чіткі абстрактні уявлення потоків інформації, які передаються між компонентами і між собою (а також характер та масштаб цих потоків) дають можливість перенаправляти їх у міру появи нових бізнес-потреб і по мірі того, як нові виробники і споживачі об'єднуються в загальній картині.
3. Протоколи XML та обміну повідомленнями, такі як SOAP, дійсно спрощують абстрагування даних та їх переміщення. Але вони також підвищують важливість того, де знаходяться дані, як вони отримують або зберігають належний контекст і як пов'язати певний синтаксис, семантику і перевірку точності з реальною інформацією, яку вони представляють.

**Процес парсера**

Однією з цілей формату XML було поліпшення форматів необроблених даних, таких як простий текст, шляхом включення докладних описів значення вмісту. Тепер, щоб мати можливість читати XML-файли, використовують синтаксичний аналізатор. Він (в основному) надає вміст документа через так званий API-інтерфейс прикладного програмування. Іншими словами, клієнтське додаток отримує доступ до вмісту XML-документа через інтерфейс замість того, щоб інтерпретувати код самостійно. Це можна продемонструвати на прикладі parser JAVA XML.

<script async="//pagead2.googlesyndication.com/pagead/js/adsbygoogle.js">   
  
<script> (adsbygoogle = window.adsbygoogle ||[]).push({});

Синтаксичний аналізатор коду аналізує правильно сформований документ, вбудований в рядок поле і передає проаналізовані дані в вихідний поле запису.

При налаштуванні аналізатора XML користувач вказує два поля. Перше містить документ, друге - призначення для проаналізованих результатів. Можна визначити елемент роздільника. Це робиться для того, щоб розділити документ на кілька значень. Якщо він не визначено, XML Parser передає весь документ в поле у вигляді карти.

При визначенні цього елемента можна використовувати його або спрощене вираз XPath. Використовують елемент, коли він знаходиться безпосередньо під кореневим вузлом і спрощене вираз XPath (для більш глибокого доступу до даних в документі XML).

<script type="text/jаvascript">   
var blockSettings = {blockId:"R-A-70350-45",renderTo:"yandex\_rtb\_R-A-70350-45",async:!0};   
if(document.cookie.indexOf("abmatch=") >= 0) blockSettings.statId = 70350;   
!function(a,b,c,d,e){a[c]=a[c]||[],a[c].push(function(){Ya.Context.AdvManager.render(blockSettings)}),e=b.getElementsByTagName("script")[0],d=b.createElement("script"),d.type="text/jаvascript",d.src="//an.yandex.ru/system/context.js",d.async=!0e.parentNode.insertBefore(d,e)}(this,this.document,"yandexContextAsyncCallbacks");

Якщо документ XML має більше одного значення, можна повернути перше значення у вигляді списку або згенерувати запис для кожного значення. При її створенні процесор включає всі інші вхідні поля в згенеровану запис. При необхідності аналізу кількох з них, процесор з-за безлічі значень включає в себе інші вхідні поля для кожної згенерованої запису.

Можна налаштувати процесор так, щоб XPath включався в кожен аналізований елемент і XML в атрибутах полів. Це також поміщає кожне простір імен в атрибут заголовка запису xmlns. Ще можна налаштувати процесор на включення атрибутів оголошень простору імен у запис (як атрибут поля). За замовчуванням він вже включає атрибути XML і оголошення простору імен у вигляді полів.

При налаштуванні аналізатора XML вказують поле для аналізу і поле виводу для використання. На панелі «Властивості» на вкладці «Загальні» налаштовують властивості, представлені в таблиці нижче.

|  |  |
| --- | --- |
| Призначення | Опис |
| Назва | Сценічний псевдонім, подібно до того, як у прикладі читання XML 1с 8 3 |
| Опис | Необов'язковий опис |
| Обов'язкові поля | Поля, які повинні включати дані для запису, переданої на сцену  Можна включити поля, які використовують сцена  Записи, які не включають усі обов'язкові поля обробляються на основі обробки помилок, налаштованої для конвеєра |
| Передумови | Умови, які повинні оцінюватися як ІСТИНА, щоб дозволити запису увійти в стадію обробки. Натискають кнопку «Додати», щоб створити додаткові попередні умови.  Записи, які не відповідають усім попереднім умовам, обробляються на основі обробки помилок, налаштованої для етапу. |
| Помилка запису | Обробка помилок запису для етапи:   * Discard - видаляє * Send to Error - відправляє в конвеєр для обробки помилок |