УДК 004

Особенности применения форматов JSON и XML при разработке корпоративных порталах

Жуков Дмитрий Сергеевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Чингалаев Сергей Алексеевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Глаголев Владимир Александрович
Приамурский государственный университет имени Шолом-Алейхема
к.г.н., доцент кафедры информационных систем, математики и правовой информатики

Аннотация
В работе рассмотрены два наиболее популярных формата обмена данными XML и JSON для серверных приложений. Рассмотрены преимущества и недостатки.
Ключевые слова: JSON, XML, передаче данных.

Features of using JSON and XML, when transferring data in corporate portals

Zhukov Dmitry Sergeevich
Sholom-Aleichem Priamursky State University
Student

Chingalaev Sergey Alekseevich
Sholom-Aleichem Priamursky State University
Student

Glagolev Vladimir Aleksandrovich
Sholom-Aleichem Priamursky State University
candidate of geographical Sciences, Associate Professor of the Department of Information Systems, Mathematics and Legal Informatics

Abstract
The paper discusses the two most popular formats for exchanging XML and JSON for server applications. Considered the advantages and disadvantages.
Keywords: JSON, XML, data transfer.

С развитием глобальной сети интернет, появились различные способы отправки и приема электронных сообщений. Создание веб-приложения подразумевает взаимодействие клиентской и серверной сторон, а именно обмен данными между ними по специализированным прикладным протоколам. Основными и самыми распространенными форматами передачи данных являются XML(e**X**tensible **M**arkup **L**anguage) и JSON(JavaScript Object Notation), каждая из которых обладает своими достоинствами и недостатками.

Целью статьи является анализ особенности применения форматов обмена данными XML и JSON для корпоративных приложений.

Задачами исследования является: обзор современных источников по данной тематике; рассмотрение прикладных средств для доступа к данным XML/JSON; разработка тестовых модулей приема данных по прикладному протоколу передачи гипертекстовых документов HTTP;

Форматы обмена данными XML и JSON для клиент-серверных приложений детально рассмотрела Е.О. Казначеева [1]. В авторской работе приведены определения и описания каждого из форматов, а также результаты анализа на основе личного опыта автора данной работы. Выделены общие черты и различия двух представленных форматов, что позволяет сделать вывод о том, какую из технологий в настоящее время можно рекомендовать большинству разработчиков программного обеспечения. В своей работе Р.Р. Гатауллин сравнил структурированного формата INI с форматами XML/JSON в контексте возможности их замены расширенной версией формата INI. Рассмотрел достоинства и недостатки конструкций присущих предлагаемой версии формата INI в сравнении с аналогичными конструкциями [2]. В статье В.В. Миронова, А.С. Гусаренко и Н.И.Юсуповой рассматрели проблему обработки документов SODB в формате JSON вместе с XML [3], так рассматриваются RESTful-сервисы как источники XML, JSON-данных использует технологию обработки XML-документов в ситуационно-ориентированных базах данных на основе динамических DOM-объектов [4]. Peter-Paul Koch рассматривает удобочитаемость кода как основной критерий анализа значимость форматов. Она является только второстепенной целью, но JSON гораздо проще воспринимается, чем XML[5].

XML

```
<font color="#069"><person></font>
   <font color="#069"><firstname></font>Subbu<font color="#069"></firstname></font>
   <font color="#069"><lastname></font>Allamaraju<font color="#069"></lastname></font>
<font color="#069"></person></font>
```

JSON

```
({
  <font color="#069">"firstName"</font> : <font color="#069">"Subbu"</font>,
  <font color="#069">"lastName"</font> : <font color="#069">"Allamaraju"</font>
});
```

Рисунок 1. Пример формата XML и JSON

Формат XML считается устаревшим, так существует набор программных интерфейсов (API) для привязки данных к XML на нескольких языках программирования. Например, на языке программирования Java можно использовать JAXB и «XmlBeans» для создания XML-ответа.

```
Person person = <font color="#069">new</font> Person();
person.setFirstName(<font color="#069">"Subbu"</font>);
person.setLastName(<font color="#069">"Allamaraju"</font>);
Marshaller marshaller = ... <font color="#008200">// Создаем обхект marshaller</font>
marshaller.marshal(person, outputStream);
```

Рисунок 2. Использованием JAXB

Все интерфейсы для создания ответа на JSON появились недавно. На JSON.org можно найти их список на различных языках.

```
Person person = <font color="#069">new</font> Person();
person.setFirstName(<font color="#069">"Subbu"</font>);
person.setLastName(<font color="#069">"Allamaraju"</font>);
writer.write(JSONObject.fromObject(person).toString());
```

Рисунок 3. Пример создания ответа при помощи «Json-lib»

В настоящее время известно больше способов генерации XML, нежели JSON. Некоторые программные интерфейсы для XML существуют достаточное время и по этой причине могут быть стабильнее при использовании для сложных приложений. Так ниже показано использование функции eval на языке программирование JavaScript.

```
var person = eval(xhr.responseText);
alert(person.firstName);
```

Рисунок 4. Использование функции eval()

Используя обычный «eval()», можно преобразовать ответ в объект JavaScript. Как только эта операция произведена, можно получить доступ к данным, используя свойства преобразованного объекта. Это наиболее изящная часть всего JSON.

Расширяемость помогает уменьшить число связей между поставщиком и получателем данных. В контексте AJAX-приложений, скрипт на стороне клиента должен быть достаточно инвариантным относительно совместимых изменениях в данных.

Для использования преимуществами расширяемости, необходимо создавать код на стороне клиента с расчетом необходимую расширяемость.

```
var xml = xhr.responseXML;
var elements = xml.getElementsByTagName("firstName");
var firstNameEl = elements[0];
var lastNameEl = firstNameEl.nextSibling;
```

Рисунок 5. Вставка элемента «middleName»

Существует и другая возможность расширить JSON-данные, она заключается в использовании вызовов функций вместе с объявлениями данных прямо в ответе.

```
alert("Hi - I'm a person");
({"firstName" : "Subbu",
  "lastName" : "Allamaraju"});
```

Рисунок 6. Вызовов функций вместе с объявлениями данных прямо в ответе

Последние версии веб-браузеров имеют встроенную поддержку JSON и способны его обрабатывать без вызова функции «eval()», приводящей к вышеописанной проблеме с безопасностью. Обработка JSON в таком случае обычно осуществляется быстрее. Так в июне 2009 года следующие браузеры имели встроенную поддержку JSON: Mozilla Firefox 3.5+; Microsoft Internet Explorer 8; Opera 10.5+; Браузеры, основанные на WebKit (Google Chrome и Apple Safari).

Пять популярных библиотек JavaScript используют встроенный JSON, в случае его доступности: jQuery; Dojo; MooTools; Yahoo! UI Library; Prototype.

Таким образом, в случае информационно-ориентированных приложений лучше использовать JSON, а не XML. XML может быть незаменимым на сервере, но с JSON определенно легче работать на клиенте.

С JSON гораздо удобнее работать на JS, чем с XML. Для других новых языков тоже есть библиотеки, поэтому приоритетнее поддержка JSON, но если API будет пользоваться большая аудитория, или будут клиенты, использующие инструменты устаревшие программные комплексы (Borland Delphi 7), то в данном случае необходима поддержка XML.

Библиографический список

1. Казначеева Е.О. Эволюция форматов обмена данными на веб-платформе на примере XML и JSON // Альманах научных работ молодых ученых университета ИТМО, 2017. С. 114-117.
2. Гатауллин Р.Р. Рассмотрение расширенной версии формата INI как замены XML / JSON // Научно-технический задел - основа эффективного инновационного развития, 2018. С. 15-20.
3. Миронов В.В., Гусаренко А.С., Юсупова Н.И. Обработка документов json с использованием ориентированных ситуацией базы данных. 2017. С.97-

103.
4. Гусаренко А.С., Миронов В.В. Использование RESTful-сервисов в ситуационно-ориентированных базах данных // Вестник Уфимского государственного авиационного технического университета. 2015. № 1 (67), С. 232-239.
5. QuirksMode.org