An Introduction to Using SOAP and REST Services for Data Retrieval

Brian R. Hall
Department of Software Technology
Champlain College
Burlington, VT, USA
hall@champlain.edu

Abstract

This technical paper illustrates the use of SOAP and REST services for Web-based data retrieval. PHP is used to initiate SOAP and REST calls to two data repositories. The data repositories utilized are the National Digital Forecast Database maintained by NOAA's National Weather Service, and Amazon's product database. The NWS example provides an introduction to using service-oriented architecture and data retrieval techniques, while the Amazon example demonstrates a more advanced scenario. In both cases, alternative data formatting options are considered. An overview of SOAP and REST literature is also provided. This paper is useful to students and novice developers seeking guidance and to computing educators seeking coursework examples.

Keywords

SOAP, REST, PHP, XML, Web-services, Service-oriented architecture, data

1 Introduction

Simple Object Access Protocol (SOAP) and Representational State Transfer (REST) are two of the most popular standards in service-oriented architecture (SOA). SOAP and REST are standards of Web-based data transfer and communication between systems. In a brief description, SOAP relies on eXtensible Markup Language (XML)-encoded messages, while REST relies on resources (in the Web sense) and hypertext transfer protocol (HTTP) commands. However, in reality both are HTTP-based data transmission, and both are commonly associated with XML, which provides a means for structuring and parsing data. Web and application developers are sometimes given a choice as to which standard to use. For example, organizations such as the National Oceanic and Atmospheric Administration (NOAA) and Amazon service both SOAP and REST requests.

The question addressed in this paper is: How can SOAP and REST be used to retrieve data? A novice programmer is likely to scour the Web for code snippets or examples. A problem with such an approach is that many examples are out-of-context, non-commented, and non-working. A second approach is to reference books such as [1, 2, 3] and try to piece together code based on the concepts and examples. A problem with the book approach is the disconnect from an ever-changing Webscape and organizational specifics. Another approach is consulting an organization's SOAP/REST documentation, which if provided may help, but can still leave a programmer guessing and looking for working examples. Programmers are likely to use a combination of these approaches and cobble code without a firm understanding of data requests using SOAP and REST.

This paper answers the posed question by presenting code and data output examples. The code examples are written in PHP. *Be aware of the line-wrapping when reviewing the code examples, each complete PHP statement ends with a semicolon.* The data output formats demonstrated in the code are: string, XML, PHP object notation, and Javascript object notation (JSON). Several output examples are provided in the Appendices. The first two code examples are at a beginner level and illustrate a simple use of SOAP and REST to retrieve data from NOAA's National Weather Service (NWS) National Digital Forecast Database (NDFD). The second two code examples are at an intermediate level and illustrate the use of SOAP and REST to retrieve data from Amazon's product database via the Product Advertising application programming interface (API). The purpose in answering the posed question is two-fold: 1) to provide students and novice developers with documented examples of SOAP and REST requests, and 2) to provide computing educators with examples they might find useful when teaching Web-services.

2 Related Literature

This review of related literature highlights some of the common themes in SOA research and provides direction for readers interested in further study of Web services, specifically SOAP and REST. One of the most common themes is "SOAP vs. REST," the purpose being to determine which is better. [4] presented a performance comparison of SOAP and REST mobile Web servers. [5] presented a performance comparison of multimedia conferencing applications using SOAP and REST. [6] presented two experiments, a client-server model implementation and a master-slave genetic algorithm. [7] compared the efficiency and scalability of SOAP and REST in meeting data transmission requirements in an interaction independence middleware framework. [8] compared SOAP and REST remote procedure calls (RPCs) in universal plug and play (UPnP) middleware. [4, 5, 6, 7, 8] each discuss pros and cons of SOAP and REST, but all five studies suggest REST offers better performance. The advantage of REST is primarily due to the XML encapsulation payload required by SOAP.

[9] proposed the SOAP service be used as part of a distributed network architecture that exploits radio frequency identification (RFID) to enable collaboration and decision-making. [10] used SOAP, along with other semantic web technology, to produce a business intelligence (BI) architecture and an analytical tool more flexible than typical BI. [11] argued that REST is the Web service approach better suited for the development of mining and learning "advocate agents" in semantic Web architecture. [12] argued that SOA can be used to support BI, either by SOAP or REST, and provided comparisons and models for integration.

[13] illustrated how both SOAP and REST can be utilized in bioinformatics Web services, and suggested that REST is better suited for data retrieval services, while SOAP is better suited for analysis services. [14] provided architectural considerations of SOAP-based services vs. RESTful services and concluded that REST is suited for basic, ad hoc integration, while SOAP is suited for the advanced requirements of enterprise computing. [15] provided a tutorial on the design and implementation of RESTful Web services, including deciding between SOAP and REST. A recurring theme throughout the literature is that SOAP is better suited for tightly-coupled environments, whereas REST is better suited for loosely-coupled environments.

[16] responsibly pointed out that the debate is somewhat misguided by the conception that SOAP and REST are rigid opposites. Both are design philosophies, but REST is fundamentally an architectural (navigational) style, while SOAP is a protocol (procedural) style used in different architectures [16]. Therefore, [16] provided a detailed breakdown of the technical debate and characteristics of SOAP and REST.

[17] and [18] exemplify two other viewpoints on SOAP and REST. [17] suggested that developers consider what SOAP and REST have in common, share service semantics, and develop specifications for uniting Web-based services. [18] offered an approach for replacing SOAP and REST in service processing.

The point of this paper is not to draw a performance comparison between SOAP and REST in order to determine which is better. (For pros and cons see the references). The author recognizes that both are useful standards, each with advantages depending on the scenario. The place this paper finds in the literature is providing readers with examples of using SOAP and REST for data retrieval.

3 National Digitial Forecast Database

The NDFD is maintained by NOAA's National Weather Service. The NDFD SOAP and REST services provide the public, government agencies, and commercial enterprises the ability to request NDFD data via the Internet. The XML dialect of data retrieved from NDFD is called Digital Weather Markup Language (DWML). NDFD data is updated no more than hourly. Information regarding NWS Web services is available at http://graphical.weather.gov.

Example 1 is a REST request to the NDFD. The REST request is structured by creating a uniform resource locator (URL) that will query the NDFD for weather data of a single point based on latitude and longitude. In Example 1, the echo command provides a clickable link that will initiate the request and return the data in XML (Appendix A).

Example 1 - REST request to NDFD

```
<?php

// Single point data
$lat = ‘‘44.48’’;
$lon = ‘‘-73.15’’;
$product = ‘‘time-series’’; // time-series or glance
$begin = ‘‘2013-02-13T12:00’’; // for time-series
$end = ‘‘2013-02-14T12:00’’; // for time-series
$Unit = ‘‘e’’;
$elements = ‘‘wx’’; // glance overrides elements

// Parameters
$String = ‘‘
lat=$lat&
lon=$lon&
product=$product&
begin=$begin&
end=$end&
Unit=$Unit&
$elements’’;

// Removes return characters in Parameters
$String = str_replace(‘‘\n’’, ‘‘’’, $String);

// Create RESTful request
$BaseUrl = ‘‘http://graphical.weather.gov/xml/sample_products/browser_interface/
ndfdXMLclient.php?’’;
$Request = $BaseUrl .  $String;

// Return data
echo ‘<a href=‘‘‘.$Request.’’’>XML</a><p>’;

?>
```

Example 2 is a SOAP request to the NDFD. The SOAP request is structured by providing a method with appropriate parameters, creating a SOAP client, and initiating the call. SOAP also requires a Web Service Description Language (WSDL) specific to the service. In Example 2, var_dump() returns the data as string output (Appendix C).

Example 2 - SOAP request to NDFD

```
<?php

// Method
$Method = ‘NDFDgen’;

// Single point data
$latitude = ‘‘44.48’’;
$longitude = ‘‘-73.15’’;
$product = ‘‘glance’’; // time-series or glance
// for time-series (commented out)
//$startTime = ‘‘2013-02-13T12:00’’;
//$endTime = ‘‘2013-02-14T12:00’’;
$Unit = ‘‘e’’;
$weatherParameters = ‘‘snow’’; // glance overrides weatherParameters

// SOAP Client
$Client = new SoapClient(‘‘http://graphical.weather.gov/xml/DWMLgen/wsdl/ndfdXML.wsdl’’);

// Parameter array
$Parameters = array(‘‘latitude’’ => $latitude,
‘‘longitude’’ => $longitude,
‘‘product’’ => $product,
‘‘startTime’’ => $startTime,
‘‘endTime’’ => $endTime,
‘‘Unit’’ => $Unit,
‘‘weatherParameters’’ => $weatherParameters);

//SOAP Call
$Results = $Client->__soapCall($Method,$Parameters);

// Return Data
var_dump($Results);

?>
```

4 Amazon Product Database

Amazon's Product Advertising API provides programmatic access to Amazon's product data. Amazon's Web service was previously called Amazon Associates Web Service (A2S) and Amazon E-Commerce Service (ECS). Data access allows developers to advertise products and monetize websites. In order to use the SOAP and REST services, a developer must join the Amazon Associates program (registration is free). Registration provides a developer with an Associate Tag and security keys necessary for making data requests. The keys are used to create an unique signature for tracking requests. In Example 3, base64 encoding helps the message survive transport across the transport protocol stack, and hash based message authentication (hash_hmac) generates a keyed hash using the security keys and the SHA256 algorithm. The signature must be included in requests. Information regarding Amazon Web services is available at http://affiliate-program.amazon.com.

Example 3 is a REST request to Amazon. The request queries Amazon's product database for item data. In Example 3, the echo command provides a clickable link that will initiate the request and return the data in XML (Appendix B, D).

Example 3 - REST request to Amazon

```
<?php

// Authentication
$AWSAccessKeyId = ‘‘YOUR_AWS_ACCESS_KEY’’;
$SecretAccessKey = ‘‘YOUR_SECRET_KEY’’;
$AssociateTag = ‘‘YOUR_ASSOCIATE_TAG’’;

// Generate timestamp
$Timestamp = gmdate(‘‘Y-m-d\TH:i:s\Z’’);
$Timestamp = str_replace(‘‘:’’, ‘‘%3A’’, $Timestamp);

// Item lookup
$ItemId = ‘‘0596007647’’; // Item desired

// Response details desired
$ResponseGroup = ‘‘ItemAttributes,Reviews’’;
$ResponseGroup = str_replace(‘‘,’’, ‘‘%2C’’,
$ResponseGroup);

// Parameters
$String = ‘‘AWSAccessKeyId=$AWSAccessKeyId&
AssociateTag=$AssociateTag&
ItemId=$ItemId&
Operation=ItemLookup&
ResponseGroup=$ResponseGroup&
Service=AWSECommerceService&
Timestamp=$Timestamp&
Version=2011-08-01’’;

// Removes return characters in Parameters
$String = str_replace(‘‘\n’’, ‘‘’’, $String);

// Request format for signature
$Prepend = ‘‘GET\necs.amazonaws.com\n/onca/xml\n’’;
$PrependString = $Prepend .  $String;

// Signature generation
$Signature = base64_encode(hash_hmac(‘‘sha256’’, $PrependString, $SecretAccessKey, True));
$Signature = str_replace(‘‘+’’, ‘‘%2B’’, $Signature);
$Signature = str_replace(‘‘=’’, ‘‘%3D’’, $Signature);
$Signature = str_replace(‘‘/’’, ‘‘%2F’’, $Signature);

// Create RESTful request
$BaseUrl = ‘‘http://ecs.amazonaws.com/onca/xml?’’;
$SignedRequest = $BaseUrl .  $String .  ‘‘&Signature=’’ .  $Signature;
```

```
// Return Data
echo ‘<a href=‘‘‘.$SignedRequest.’’’>XML</a><p>’;
$XML = simplexml_load_file($SignedRequest);
print_r ($XML);

?>
```

Example 4 is a SOAP request to Amazon. Example 4 illustrates how to provide the PHP script with data entered in an HTML form. For simplicity, the form only prompts for the ItemID, but could be modified to prompt for all the variables. Example 4 also illustrates PHP object notation and JSON output options (Appendix E).

Example 4 - SOAP request to Amazon - HTML file

```
<!DOCTYPE html>
<form action = "soap.php" method = "post">
Item ID: <input type = "text" name = "search"><br>
<input type = "submit" value = "submit"
name = "submit">
</form>
</html>
```

Example 4 - SOAP request to Amazon - PHP file

```
<?php

// Verify search term
echo ‘‘Your search:  {$_POST[‘search’]} <br />’’;

// Authentication
$AWSAccessKeyId = ‘‘YOUR_AWS_ACCESS_KEY’’;
$SecretAccessKey = ‘‘YOUR_SECRET_KEY’’;
$AssociateTag = ‘‘YOUR_ASSOCIATE_TAG’’;

// Generate timestamp
$Timestamp = gmdate(‘‘Y-m-d\TH:i:s\Z’’);

// Method
$Method = ‘ItemLookup’;

// Generate prepend string
$PrependString = $Method.$Timestamp;

// Generate Signature
$Signature = base64_encode(hash_hmac(‘‘sha256’’, $PrependString, $SecretAccessKey, True));

// SOAP Client
$Client = new SoapClient(‘‘http://webservices.amazon.com/AWSECommerceService/2011-08-01/
US/AWSECommerceService.wsdl’’);
```

```
// Set headers
$HeaderSet = array(
‘AWSAccessKeyId’ => $AWSAccessKeyId,
‘AssociateTag’ => $AssociateTag,
‘Timestamp’ => $Timestamp,
‘Signature’=> $Signature);
$headers = array();
foreach ($HeaderSet as $k => $v) {
   array_push($headers, new SoapHeader(‘http://security.amazonaws.com/doc/
   2007-01-01/’, $k, $v)); }
$Client->__setSoapHeaders($headers);

// Request and Parameter arrays
$Request = array(‘Operation’=> $Method,
‘ItemId’ => $_POST[‘search’],
‘ResponseGroup’=> ‘Medium’ );
$Parameters = array(
‘‘Service’’ => ‘AWSECommerceService’,
‘‘AWSAccessKeyId’’ => $AWSAccessKeyId,
‘‘AssociateTag’’ => $AssociateTag,
‘‘Operation’’ => $Method,
‘‘Request’’ => $Request);

//SOAP Call
$Results = $Client->__soapCall($Method, array($Parameters));

// Return Data
echo json_encode($Results); // JSON ouptut
// var_dump($Results); // PHP object output

?>
```

5 Conclusion

This paper presented four PHP code examples using the SOA services SOAP and REST for data retrieval. The data retrieved was formatted using XML, JSON, PHP object notation, and string output. The databases used to illustrate the concepts were the NDFD and Amazon's product database. The purpose of this paper was not to compare the performance of SOAP vs. REST, but to demonstrate their utility in data retrieval. This paper provides students and novice developers with documented examples of SOAP and REST requests, and provides computing educators with examples they might find useful when teaching Web-services.

References

[1] E. Cerami, *Web services essentials.* O'Reilly Media, Feb. 2002.

[2] J. Snell, D. Tidwell, and P. Kulchenko, *Programming web services with SOAP*, 1st ed. O'Reilly Media, Dec. 2001.

[3] L. Richardson and S. Ruby, *Restful web services*, 1st ed. O'Reilly Media, May 2007.

[4] F. Aijaz, M. Chaudhary, and B. Walke, “Performance comparison of a SOAP and REST mobile web server,” in *Proceedings of the 3rd International Conference on Open Source Systems and Technologies*, 2009.

[5] F. Belqasmi, J. Singh, S. Y. B. Melhem, and R. H. Glitho, “SOAP-based vs. RESTful web services: A case study for multimedia conferencing,” *IEEE Internet Computing*, vol. 16, no. 4, pp. 54–63, 2012.

[6] P. A. Castillo, J. L. Bernier, M. G. Arenas, J. J. Merelo, and P. Garcia-Sanchez, “SOAP vs REST: Comparing a master-slave GA implementation,” *ArXiv e-prints*, no. 1105.4978, May 2011. [Online]. Available: http://arxiv.org/abs/1105.4978

[7] G. Mulligan and D. Gracanin, “A comparison of SOAP and REST implementations of a service based interaction independence middleware framework,” in *Proceedings of the 2009 Winter Simulation Conference*. IEEE, 2009, pp. 1423–1432.

[8] J. Newmarch, “A RESTful approach: Clean UPnP without SOAP,” in *Second IEEE Consumer Communications and Networking Conference*. IEEE, 2005, pp. 134–138.

[9] G. Lekakos, “Exploiting RFID digital information in enterprise collaboration,” *Industrial Management and Data Systems*, vol. 107, no. 8, pp. 1110–1122, 2007.

[10] D. Sell, L. Cabral, E. Motta, J. Domingue, and R. Pacheco, “Adding semantics to business intelligence,” in *Proceedings of the 16th International Workshop on Database and Expert Systems*. IEEE, 2005, pp. 543–547.

[11] W. Ketter, A. Batchu, G. Berosik, and D. McCreary, “A semantic web architecture for advocate agents to determine preferences and facilitate decision making,” in *Proceedings of the 10th International Conference on Electronic Commerce*. ACM, 2008.

[12] P. Wik, “Service-oriented architecture and business intelligence,” *Service Technology Magazine*, no. LIII, pp. 1–18, 2011.

[13] T. Katayama, M. Nakao, and T. Takagi, “TogoWS: integrated SOAP and REST APIs for interoperable bioinformatics web services,” *Nucleic Acids Research*, vol. 38, no. 2, pp. W706–W711, 2010.

[14] C. Pautasso, O. Zimmermann, and F. Leymann, “RESTful web services vs. ”Big” web services: Making the right architectural decision,” in *Proceedings of the 17th International Conference on World Wide Web*. ACM, 2008, pp. 805–814.

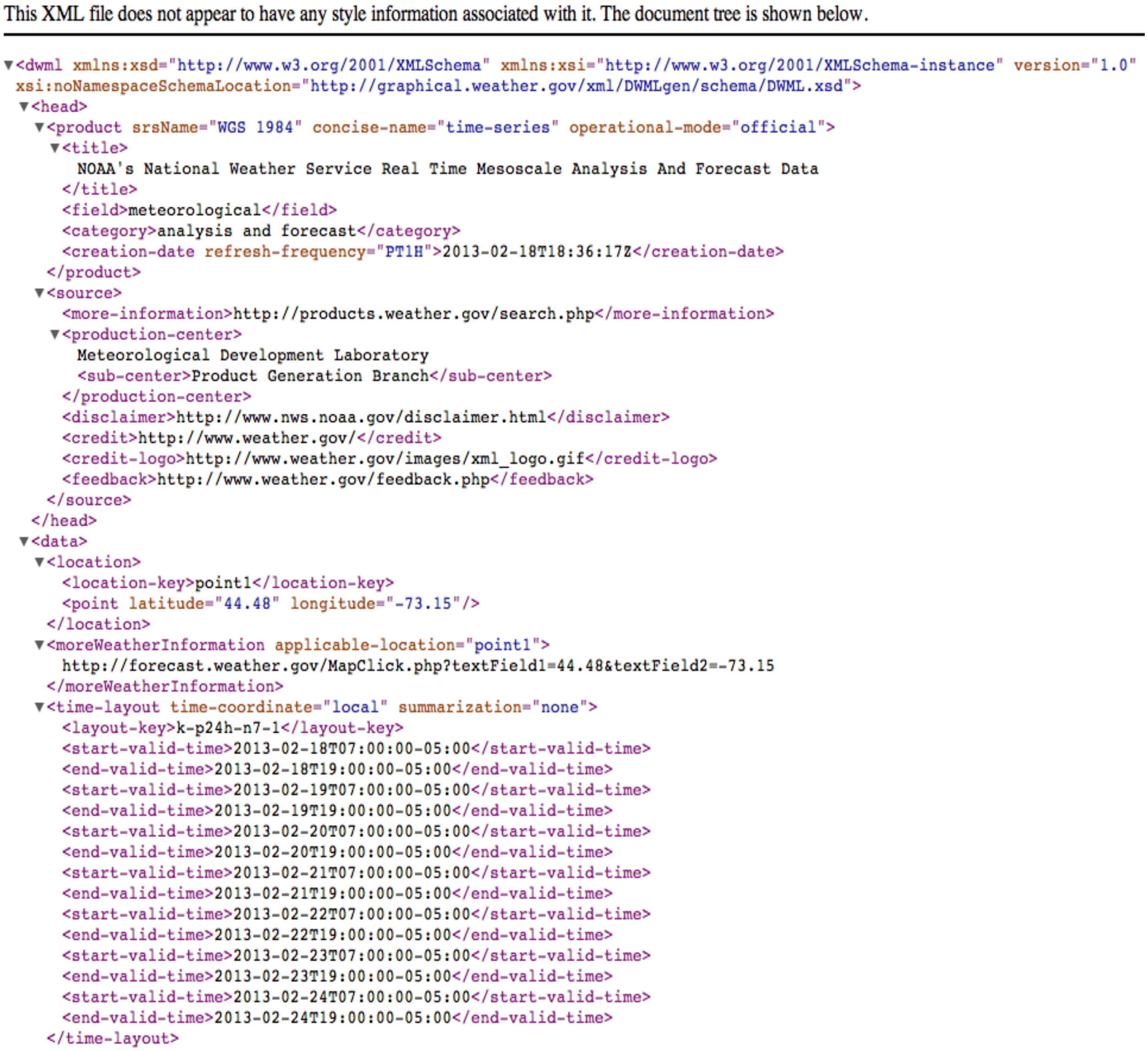
[15] C. Pautasso and E. Wilde, “RESTful web services: Principles, patterns, emerging technologies,” in *Proceedings of the 19th International Conference on World Wide Web*. ACM, 2010, pp. 1359–1360.

[16] M. zur Muehlen, J. V. Nickerson, and K. D. Swenson, “Developing web services choreography standards: The case of REST vs. SOAP,” *Decision Support Systems*, vol. 40, no. 1, pp. 9–29, 2005.

[17] X. Shi, “Sharing service semantics using SOAP-based and REST web services,” *IT Professional*, vol. 8, no. 2, pp. 18–24, 2006.

[18] T. De Giorgio, G. Ripa, and M. Zuccala, “An approach to enable replacement of SOAP services and REST services in lightweight processes,” *Current Trends in Web Engineering*, pp. 338–346, 2010.

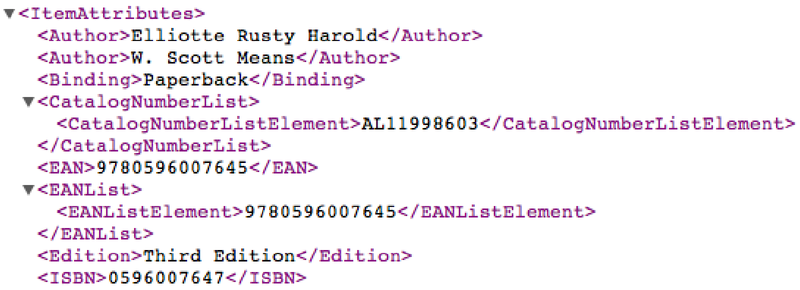
A REST - Portion of NDFD XML output



This XML file does not appear to have any style information associated with it. The document tree is shown below.

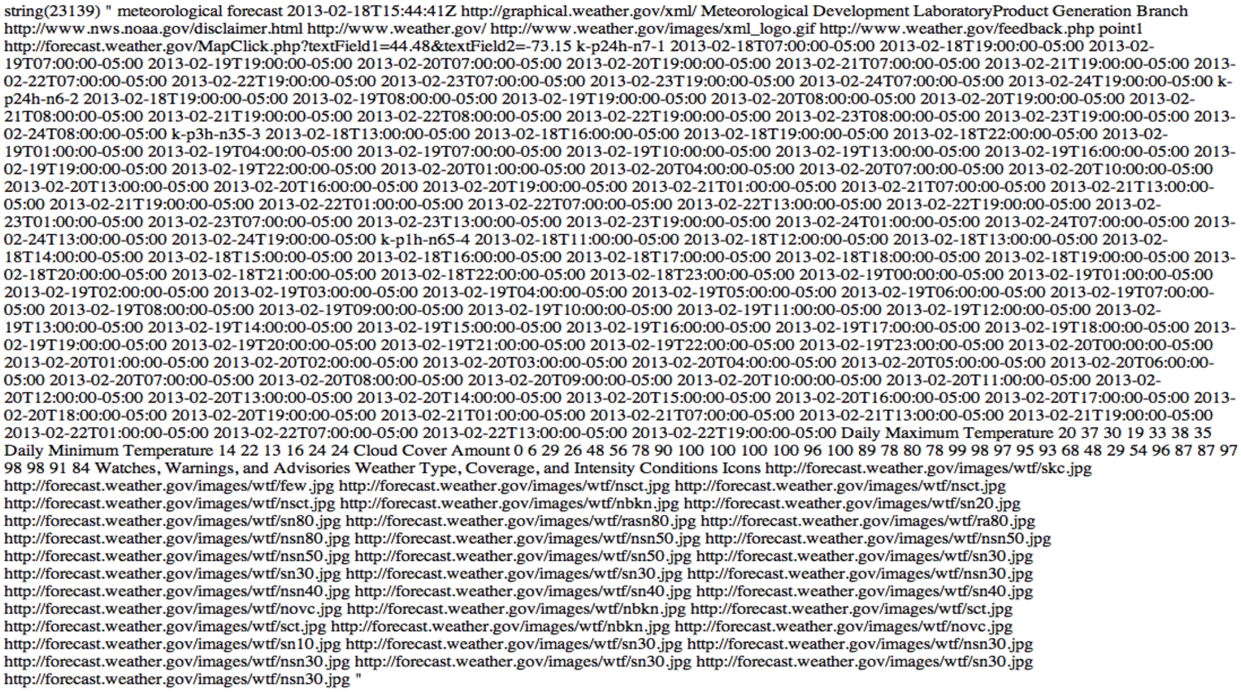
```
▼<dwml xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0"
 xsi:noNamespaceSchemaLocation="http://graphical.weather.gov/xml/DWMLgen/schema/DWML.xsd">
 ▼<head>
  ▼<product srsName="WGS 1984" concise-name="time-series" operational-mode="official">
   ▼<title>
      NOAA's National Weather Service Real Time Mesoscale Analysis And Forecast Data
    </title>
    <field>meteorological</field>
    <category>analysis and forecast</category>
    <creation-date refresh-frequency="PT1H">2013-02-18T18:36:17Z</creation-date>
  </product>
  ▼<source>
    <more-information>http://products.weather.gov/search.php</more-information>
   ▼<production-center>
      Meteorological Development Laboratory
      <sub-center>Product Generation Branch</sub-center>
    </production-center>
    <disclaimer>http://www.nws.noaa.gov/disclaimer.html</disclaimer>
    <credit>http://www.weather.gov/</credit>
    <credit-logo>http://www.weather.gov/images/xml_logo.gif</credit-logo>
    <feedback>http://www.weather.gov/feedback.php</feedback>
  </source>
 </head>
 ▼<data>
  ▼<location>
    <location-key>point1</location-key>
    <point latitude="44.48" longitude="-73.15"/>
  </location>
  ▼<moreWeatherInformation applicable-location="point1">
    http://forecast.weather.gov/MapClick.php?textField1=44.48&textField2=-73.15
  </moreWeatherInformation>
  ▼<time-layout time-coordinate="local" summarization="none">
    <layout-key>k-p24h-n7-1</layout-key>
    <start-valid-time>2013-02-18T07:00:00-05:00</start-valid-time>
    <end-valid-time>2013-02-18T19:00:00-05:00</end-valid-time>
    <start-valid-time>2013-02-19T07:00:00-05:00</start-valid-time>
    <end-valid-time>2013-02-19T19:00:00-05:00</end-valid-time>
    <start-valid-time>2013-02-20T07:00:00-05:00</start-valid-time>
    <end-valid-time>2013-02-20T19:00:00-05:00</end-valid-time>
    <start-valid-time>2013-02-21T07:00:00-05:00</start-valid-time>
    <end-valid-time>2013-02-21T19:00:00-05:00</end-valid-time>
    <start-valid-time>2013-02-22T07:00:00-05:00</start-valid-time>
    <end-valid-time>2013-02-22T19:00:00-05:00</end-valid-time>
    <start-valid-time>2013-02-23T07:00:00-05:00</start-valid-time>
    <end-valid-time>2013-02-23T19:00:00-05:00</end-valid-time>
    <start-valid-time>2013-02-24T07:00:00-05:00</start-valid-time>
    <end-valid-time>2013-02-24T19:00:00-05:00</end-valid-time>
  </time-layout>
```

B REST - Portion of Amazon XML output



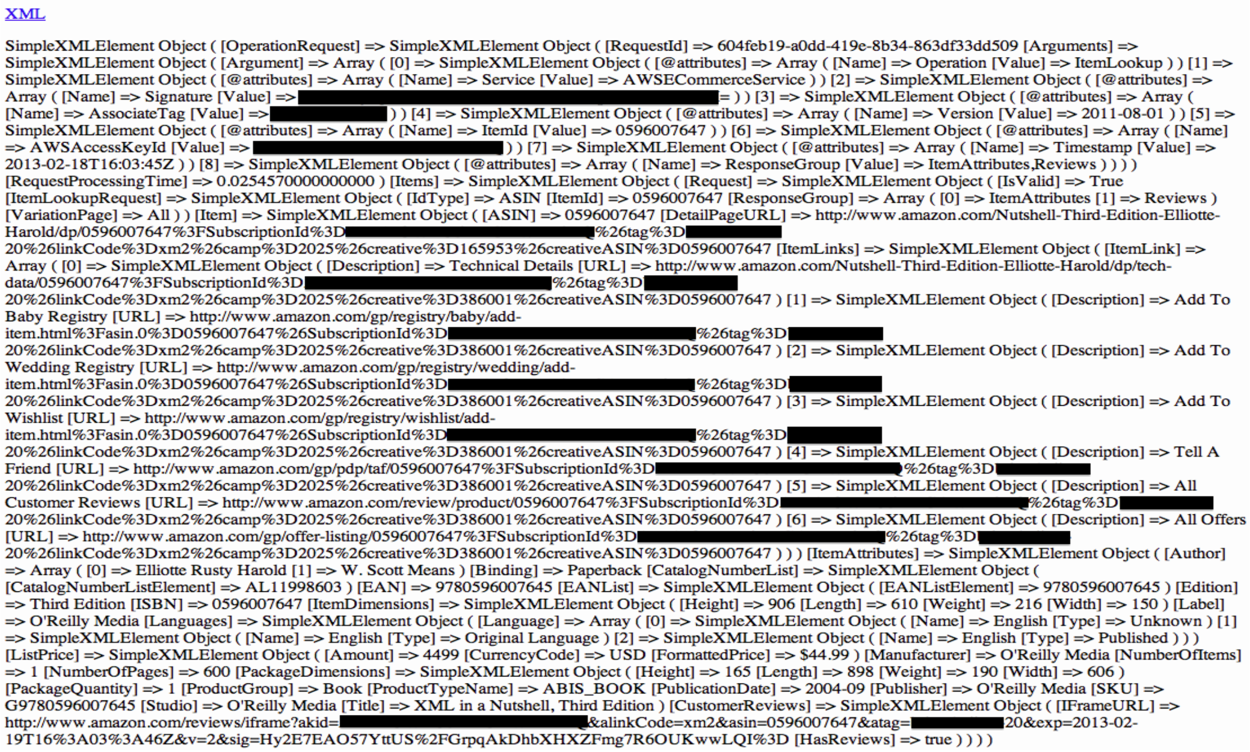
```
▼<ItemAttributes>
   <Author>Elliotte Rusty Harold</Author>
   <Author>W. Scott Means</Author>
   <Binding>Paperback</Binding>
  ▼<CatalogNumberList>
     <CatalogNumberListElement>AL11998603</CatalogNumberListElement>
   </CatalogNumberList>
   <EAN>9780596007645</EAN>
  ▼<EANList>
     <EANListElement>9780596007645</EANListElement>
   </EANList>
   <Edition>Third Edition</Edition>
   <ISBN>0596007647</ISBN>
```

C SOAP - NDFD string output



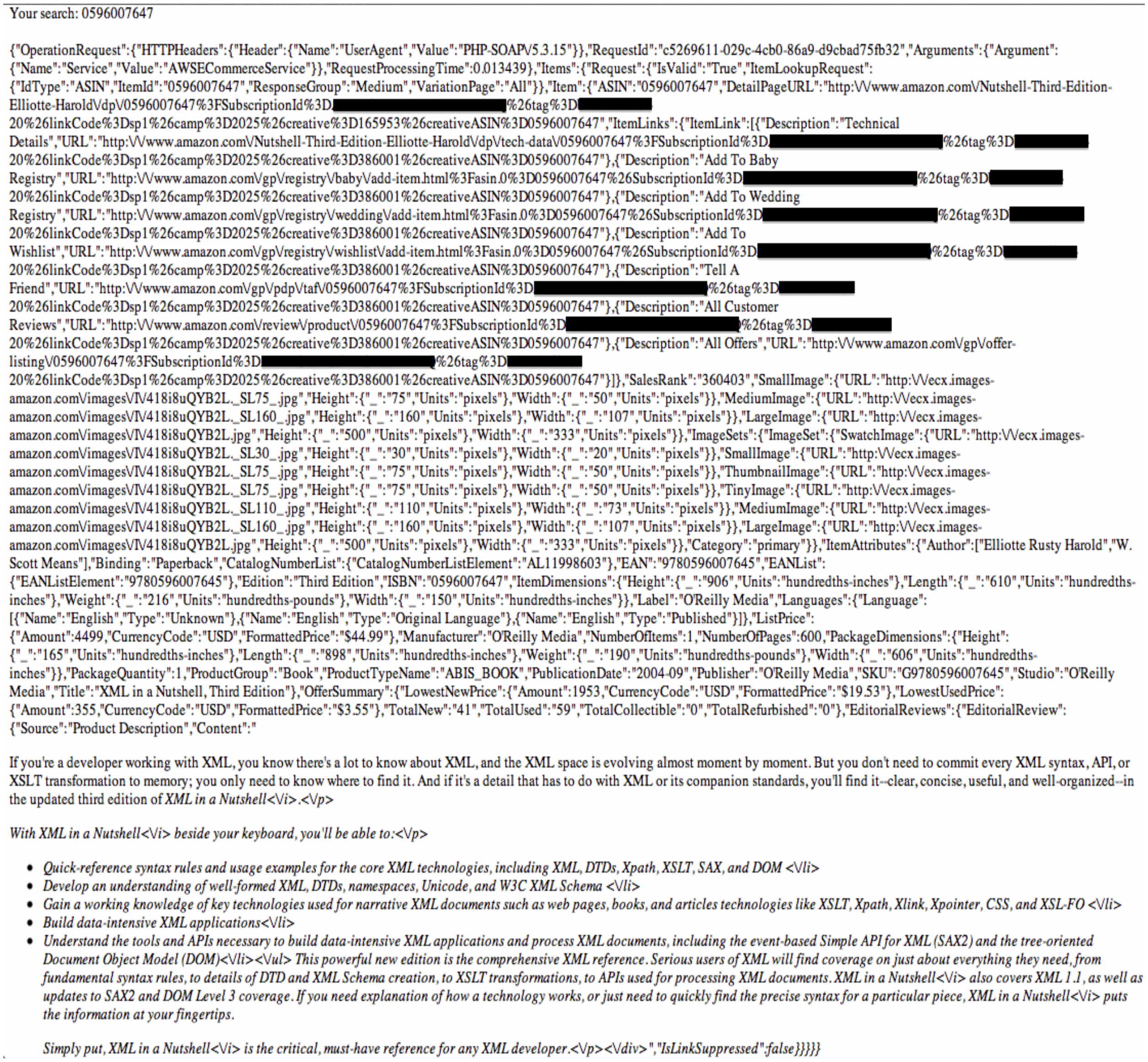
```
string(23139) " meteorological forecast 2013-02-18T15:44:41Z http://graphical.weather.gov/xml/ Meteorological Development LaboratoryProduct Generation Branch
http://www.nws.noaa.gov/disclaimer.html http://www.weather.gov/ http://www.weather.gov/images/xml_logo.gif http://www.weather.gov/feedback.php point1
http://forecast.weather.gov/MapClick.php?textField1=44.48&textField2=-73.15 k-p24h-n7-1 2013-02-18T07:00:00-05:00 2013-02-18T19:00:00-05:00 2013-02-
19T07:00:00-05:00 2013-02-19T19:00:00-05:00 2013-02-20T07:00:00-05:00 2013-02-20T19:00:00-05:00 2013-02-21T07:00:00-05:00 2013-02-21T19:00:00-05:00 2013-
02-22T07:00:00-05:00 2013-02-22T19:00:00-05:00 2013-02-23T07:00:00-05:00 2013-02-23T19:00:00-05:00 2013-02-24T07:00:00-05:00 2013-02-24T19:00:00-05:00 k-
p24h-n6-2 2013-02-18T19:00:00-05:00 2013-02-19T08:00:00-05:00 2013-02-19T19:00:00-05:00 2013-02-20T08:00:00-05:00 2013-02-20T19:00:00-05:00 2013-02-
21T08:00:00-05:00 2013-02-21T19:00:00-05:00 2013-02-22T08:00:00-05:00 2013-02-22T19:00:00-05:00 2013-02-23T08:00:00-05:00 2013-02-23T19:00:00-05:00 2013-
02-24T08:00:00-05:00 k-p3h-n35-3 2013-02-18T13:00:00-05:00 2013-02-18T16:00:00-05:00 2013-02-18T19:00:00-05:00 2013-02-18T22:00:00-05:00 2013-02-
19T01:00:00-05:00 2013-02-19T04:00:00-05:00 2013-02-19T07:00:00-05:00 2013-02-19T10:00:00-05:00 2013-02-19T13:00:00-05:00 2013-02-19T16:00:00-05:00 2013-
02-19T19:00:00-05:00 2013-02-19T22:00:00-05:00 2013-02-20T01:00:00-05:00 2013-02-20T04:00:00-05:00 2013-02-20T07:00:00-05:00 2013-02-20T10:00:00-05:00
2013-02-20T13:00:00-05:00 2013-02-20T16:00:00-05:00 2013-02-20T19:00:00-05:00 2013-02-21T01:00:00-05:00 2013-02-21T07:00:00-05:00 2013-02-21T13:00:00-
05:00 2013-02-21T19:00:00-05:00 2013-02-22T01:00:00-05:00 2013-02-22T07:00:00-05:00 2013-02-22T13:00:00-05:00 2013-02-22T19:00:00-05:00 2013-02-
23T01:00:00-05:00 2013-02-23T07:00:00-05:00 2013-02-23T13:00:00-05:00 2013-02-23T19:00:00-05:00 2013-02-24T01:00:00-05:00 2013-02-24T07:00:00-05:00 2013-
02-24T13:00:00-05:00 2013-02-24T19:00:00-05:00 k-p1h-n65-4 2013-02-18T11:00:00-05:00 2013-02-18T12:00:00-05:00 2013-02-18T13:00:00-05:00 2013-02-
18T14:00:00-05:00 2013-02-18T15:00:00-05:00 2013-02-18T16:00:00-05:00 2013-02-18T17:00:00-05:00 2013-02-18T18:00:00-05:00 2013-02-18T19:00:00-05:00 2013-
02-18T20:00:00-05:00 2013-02-18T21:00:00-05:00 2013-02-18T22:00:00-05:00 2013-02-18T23:00:00-05:00 2013-02-19T00:00:00-05:00 2013-02-19T01:00:00-05:00
2013-02-19T02:00:00-05:00 2013-02-19T03:00:00-05:00 2013-02-19T04:00:00-05:00 2013-02-19T05:00:00-05:00 2013-02-19T06:00:00-05:00 2013-02-19T07:00:00-
05:00 2013-02-19T08:00:00-05:00 2013-02-19T09:00:00-05:00 2013-02-19T10:00:00-05:00 2013-02-19T11:00:00-05:00 2013-02-19T12:00:00-05:00 2013-02-
19T13:00:00-05:00 2013-02-19T14:00:00-05:00 2013-02-19T15:00:00-05:00 2013-02-19T16:00:00-05:00 2013-02-19T17:00:00-05:00 2013-02-19T18:00:00-05:00 2013-
02-19T19:00:00-05:00 2013-02-19T20:00:00-05:00 2013-02-19T21:00:00-05:00 2013-02-19T22:00:00-05:00 2013-02-19T23:00:00-05:00 2013-02-20T00:00:00-05:00
2013-02-20T01:00:00-05:00 2013-02-20T02:00:00-05:00 2013-02-20T03:00:00-05:00 2013-02-20T04:00:00-05:00 2013-02-20T05:00:00-05:00 2013-02-20T06:00:00-
05:00 2013-02-20T07:00:00-05:00 2013-02-20T08:00:00-05:00 2013-02-20T09:00:00-05:00 2013-02-20T10:00:00-05:00 2013-02-20T11:00:00-05:00 2013-02-
20T12:00:00-05:00 2013-02-20T13:00:00-05:00 2013-02-20T14:00:00-05:00 2013-02-20T15:00:00-05:00 2013-02-20T16:00:00-05:00 2013-02-20T17:00:00-05:00 2013-
02-20T18:00:00-05:00 2013-02-20T19:00:00-05:00 2013-02-21T01:00:00-05:00 2013-02-21T07:00:00-05:00 2013-02-21T13:00:00-05:00 2013-02-21T19:00:00-05:00
2013-02-22T01:00:00-05:00 2013-02-22T07:00:00-05:00 2013-02-22T13:00:00-05:00 2013-02-22T19:00:00-05:00 Daily Maximum Temperature 20 37 30 19 33 38 35
Daily Minimum Temperature 14 22 13 16 24 24 Cloud Cover Amount 0 6 29 26 48 56 78 90 100 100 100 100 96 100 89 78 80 78 99 98 97 95 93 68 48 29 54 96 87 87 97
98 98 91 84 Watches, Warnings, and Advisories Weather Type, Coverage, and Intensity Conditions Icons http://forecast.weather.gov/images/wtf/skc.jpg
http://forecast.weather.gov/images/wtf/few.jpg http://forecast.weather.gov/images/wtf/nsct.jpg http://forecast.weather.gov/images/wtf/nsct.jpg
http://forecast.weather.gov/images/wtf/nsct.jpg http://forecast.weather.gov/images/wtf/nbkn.jpg http://forecast.weather.gov/images/wtf/sn20.jpg
http://forecast.weather.gov/images/wtf/sn80.jpg http://forecast.weather.gov/images/wtf/rasn80.jpg http://forecast.weather.gov/images/wtf/ra80.jpg
http://forecast.weather.gov/images/wtf/nsn80.jpg http://forecast.weather.gov/images/wtf/nsn50.jpg http://forecast.weather.gov/images/wtf/nsn50.jpg
http://forecast.weather.gov/images/wtf/nsn50.jpg http://forecast.weather.gov/images/wtf/sn50.jpg http://forecast.weather.gov/images/wtf/sn30.jpg
http://forecast.weather.gov/images/wtf/sn30.jpg http://forecast.weather.gov/images/wtf/sn30.jpg http://forecast.weather.gov/images/wtf/nsn30.jpg
http://forecast.weather.gov/images/wtf/nsn40.jpg http://forecast.weather.gov/images/wtf/sn40.jpg http://forecast.weather.gov/images/wtf/sn40.jpg
http://forecast.weather.gov/images/wtf/novc.jpg http://forecast.weather.gov/images/wtf/nbkn.jpg http://forecast.weather.gov/images/wtf/sct.jpg
http://forecast.weather.gov/images/wtf/sct.jpg http://forecast.weather.gov/images/wtf/nbkn.jpg http://forecast.weather.gov/images/wtf/novc.jpg
http://forecast.weather.gov/images/wtf/sn10.jpg http://forecast.weather.gov/images/wtf/sn30.jpg http://forecast.weather.gov/images/wtf/nsn30.jpg
http://forecast.weather.gov/images/wtf/nsn30.jpg http://forecast.weather.gov/images/wtf/sn30.jpg http://forecast.weather.gov/images/wtf/sn30.jpg
http://forecast.weather.gov/images/wtf/nsn30.jpg "
```

D REST - Amazon XML/object output



```
SimpleXMLElement Object ( [OperationRequest] => SimpleXMLElement Object ( [RequestId] => 604feb19-a0dd-419e-8b34-863df33dd509 [Arguments] =>
SimpleXMLElement Object ( [Argument] => Array ( [0] => SimpleXMLElement Object ( [@attributes] => Array ( [Name] => Operation [Value] => ItemLookup ) ) [1] =>
SimpleXMLElement Object ( [@attributes] => Array ( [Name] => Service [Value] => AWSECommerceService ) ) [2] => SimpleXMLElement Object ( [@attributes] =>
Array ( [Name] => Signature [Value] =>[redacted]= ) ) [3] => SimpleXMLElement Object ( [@attributes] => Array (
[Name] => AssociateTag [Value] =>[redacted] ) ) [4] => SimpleXMLElement Object ( [@attributes] => Array ( [Name] => Version [Value] => 2011-08-01 ) ) [5] =>
SimpleXMLElement Object ( [@attributes] => Array ( [Name] => ItemId [Value] => 0596007647 ) ) [6] => SimpleXMLElement Object ( [@attributes] => Array ( [Name]
=> AWSAccessKeyId [Value] =>[redacted] ) ) [7] => SimpleXMLElement Object ( [@attributes] => Array ( [Name] => Timestamp [Value] =>
2013-02-18T16:03:45Z ) ) [8] => SimpleXMLElement Object ( [@attributes] => Array ( [Name] => ResponseGroup [Value] => ItemAttributes,Reviews ) ) ) )
[RequestProcessingTime] => 0.0254570000000000 ) [Items] => SimpleXMLElement Object ( [Request] => SimpleXMLElement Object ( [IsValid] => True
[ItemLookupRequest] => SimpleXMLElement Object ( [IdType] => ASIN [ItemId] => 0596007647 [ResponseGroup] => Array ( [0] => ItemAttributes [1] => Reviews )
[VariationPage] => All ) ) [Item] => SimpleXMLElement Object ( [ASIN] => 0596007647 [DetailPageURL] => http://www.amazon.com/Nutshell-Third-Edition-Elliotte-
Harold/dp/0596007647%3FSubscriptionId%3D[redacted]%26tag%3D[redacted]
20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0596007647 [ItemLinks] => SimpleXMLElement Object ( [ItemLink] =>
Array ( [0] => SimpleXMLElement Object ( [Description] => Technical Details [URL] => http://www.amazon.com/Nutshell-Third-Edition-Elliotte-Harold/dp/tech-
data/0596007647%3FSubscriptionId%3D[redacted]%26tag%3D[redacted]
20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D386001%26creativeASIN%3D0596007647 ) [1] => SimpleXMLElement Object ( [Description] => Add To
Baby Registry [URL] => http://www.amazon.com/gp/registry/baby/add-
item.html%3Fasin.0%3D0596007647%26SubscriptionId%3D[redacted]%26tag%3D[redacted]
20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D386001%26creativeASIN%3D0596007647 ) [2] => SimpleXMLElement Object ( [Description] => Add To
Wedding Registry [URL] => http://www.amazon.com/gp/registry/wedding/add-
item.html%3Fasin.0%3D0596007647%26SubscriptionId%3D[redacted]%26tag%3D[redacted]
20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D386001%26creativeASIN%3D0596007647 ) [3] => SimpleXMLElement Object ( [Description] => Add To
Wishlist [URL] => http://www.amazon.com/gp/registry/wishlist/add-
item.html%3Fasin.0%3D0596007647%26SubscriptionId%3D[redacted]%26tag%3D[redacted]
20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D386001%26creativeASIN%3D0596007647 ) [4] => SimpleXMLElement Object ( [Description] => Tell A
Friend [URL] => http://www.amazon.com/gp/pdp/taf/0596007647%3FSubscriptionId%3D[redacted]%26tag%3D[redacted]
20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D386001%26creativeASIN%3D0596007647 ) [5] => SimpleXMLElement Object ( [Description] => All
Customer Reviews [URL] => http://www.amazon.com/review/product/0596007647%3FSubscriptionId%3D[redacted]%26tag%3D[redacted]
20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D386001%26creativeASIN%3D0596007647 ) [6] => SimpleXMLElement Object ( [Description] => All Offers
[URL] => http://www.amazon.com/gp/offer-listing/0596007647%3FSubscriptionId%3D[redacted]%26tag%3D[redacted]
20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D386001%26creativeASIN%3D0596007647 ) ) ) [ItemAttributes] => SimpleXMLElement Object ( [Author]
=> Array ( [0] => Elliotte Rusty Harold [1] => W. Scott Means ) [Binding] => Paperback [CatalogNumberList] => SimpleXMLElement Object (
[CatalogNumberListElement] => AL11998603 ) [EAN] => 9780596007645 [EANList] => SimpleXMLElement Object ( [EANListElement] => 9780596007645 ) [Edition]
=> Third Edition [ISBN] => 0596007647 [ItemDimensions] => SimpleXMLElement Object ( [Height] => 906 [Length] => 610 [Weight] => 216 [Width] => 150 ) [Label]
=> O'Reilly Media [Languages] => SimpleXMLElement Object ( [Language] => Array ( [0] => SimpleXMLElement Object ( [Name] => English [Type] => Unknown ) [1]
=> SimpleXMLElement Object ( [Name] => English [Type] => Original Language ) [2] => SimpleXMLElement Object ( [Name] => English [Type] => Published ) ) )
[ListPrice] => SimpleXMLElement Object ( [Amount] => 4499 [CurrencyCode] => USD [FormattedPrice] => $44.99 ) [Manufacturer] => O'Reilly Media [NumberOfItems]
=> 1 [NumberOfPages] => 600 [PackageDimensions] => SimpleXMLElement Object ( [Height] => 165 [Length] => 898 [Weight] => 190 [Width] => 606 )
[PackageQuantity] => 1 [ProductGroup] => Book [ProductTypeName] => ABIS_BOOK [PublicationDate] => 2004-09 [Publisher] => O'Reilly Media [SKU] =>
G9780596007645 [Studio] => O'Reilly Media [Title] => XML in a Nutshell, Third Edition ) [CustomerReviews] => SimpleXMLElement Object ( [IFrameURL] =>
http://www.amazon.com/reviews/iframe?akid=[redacted]&alinkCode=xm2&asin=0596007647&atag=[redacted]20&exp=2013-02-
19T16%3A03%3A46Z&v=2&sig=Hy2E7EAO57YttUS%2FGrpqAkDhbXHXZFmg7R6OUKwwLQI%3D [HasReviews] => true ) ) ) )
```

E SOAP - Amazon JSON output



Your search: 0596007647

{"OperationRequest":{"HTTPHeaders":{"Header":{"Name":"UserAgent","Value":"PHP-SOAPV5.3.15"}},"RequestId":"c5269611-029c-4cb0-86a9-d9cbad75fb32","Arguments":{"Argument":
{"Name":"Service","Value":"AWSECommerceService"}},"RequestProcessingTime":0.013439},"Items":{"Request":{"IsValid":"True","ItemLookupRequest":
{"IdType":"ASIN","ItemId":"0596007647","ResponseGroup":"Medium","VariationPage":"All"}},"Item":{"ASIN":"0596007647","DetailPageURL":"http:\/\/www.amazon.com\/Nutshell-Third-Edition-Elliotte-Harold\/dp\/0596007647%3FSubscriptionId%3D[redacted]%26tag%3D[redacted]
20%26linkCode%3Dsp1%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0596007647","ItemLinks":{"ItemLink":[{"Description":"Technical
Details","URL":"http:\/\/www.amazon.com\/Nutshell-Third-Edition-Elliotte-Harold\/dp\/tech-data\/0596007647%3FSubscriptionId%3D[redacted]%26tag%3D[redacted]
20%26linkCode%3Dsp1%26camp%3D2025%26creative%3D386001%26creativeASIN%3D0596007647"},{"Description":"Add To Baby
Registry","URL":"http:\/\/www.amazon.com\/gp\/registry\/baby\/add-item.html%3Fasin.0%3D0596007647%26SubscriptionId%3D[redacted]%26tag%3D[redacted]
20%26linkCode%3Dsp1%26camp%3D2025%26creative%3D386001%26creativeASIN%3D0596007647"},{"Description":"Add To Wedding
Registry","URL":"http:\/\/www.amazon.com\/gp\/registry\/wedding\/add-item.html%3Fasin.0%3D0596007647%26SubscriptionId%3D[redacted]%26tag%3D[redacted]
20%26linkCode%3Dsp1%26camp%3D2025%26creative%3D386001%26creativeASIN%3D0596007647"},{"Description":"Add To
Wishlist","URL":"http:\/\/www.amazon.com\/gp\/registry\/wishlist\/add-item.html%3Fasin.0%3D0596007647%26SubscriptionId%3D[redacted]%26tag%3D[redacted]
20%26linkCode%3Dsp1%26camp%3D2025%26creative%3D386001%26creativeASIN%3D0596007647"},{"Description":"Tell A
Friend","URL":"http:\/\/www.amazon.com\/gp\/pdp\/taf\/0596007647%3FSubscriptionId%3D[redacted]%26tag%3D[redacted]
20%26linkCode%3Dsp1%26camp%3D2025%26creative%3D386001%26creativeASIN%3D0596007647"},{"Description":"All Customer
Reviews","URL":"http:\/\/www.amazon.com\/review\/product\/0596007647%3FSubscriptionId%3D[redacted]%26tag%3D[redacted]
20%26linkCode%3Dsp1%26camp%3D2025%26creative%3D386001%26creativeASIN%3D0596007647"},{"Description":"All Offers","URL":"http:\/\/www.amazon.com\/gp\/offer-listing\/0596007647%3FSubscriptionId%3D[redacted]%26tag%3D[redacted]
20%26linkCode%3Dsp1%26camp%3D2025%26creative%3D386001%26creativeASIN%3D0596007647"}]},"SalesRank":"360403","SmallImage":{"URL":"http:\/\/ecx.images-amazon.com\/images\/I\/418i8uQYB2L._SL75_.jpg","Height":{"_":"75","Units":"pixels"},"Width":{"_":"50","Units":"pixels"}},"MediumImage":{"URL":"http:\/\/ecx.images-amazon.com\/images\/I\/418i8uQYB2L._SL160_.jpg","Height":{"_":"160","Units":"pixels"},"Width":{"_":"107","Units":"pixels"}},"LargeImage":{"URL":"http:\/\/ecx.images-amazon.com\/images\/I\/418i8uQYB2L.jpg","Height":{"_":"500","Units":"pixels"},"Width":{"_":"333","Units":"pixels"}},"ImageSets":{"ImageSet":{"SwatchImage":{"URL":"http:\/\/ecx.images-amazon.com\/images\/I\/418i8uQYB2L._SL30_.jpg","Height":{"_":"30","Units":"pixels"},"Width":{"_":"20","Units":"pixels"}},"SmallImage":{"URL":"http:\/\/ecx.images-amazon.com\/images\/I\/418i8uQYB2L._SL75_.jpg","Height":{"_":"75","Units":"pixels"},"Width":{"_":"50","Units":"pixels"}},"ThumbnailImage":{"URL":"http:\/\/ecx.images-amazon.com\/images\/I\/418i8uQYB2L._SL75_.jpg","Height":{"_":"75","Units":"pixels"},"Width":{"_":"50","Units":"pixels"}},"TinyImage":{"URL":"http:\/\/ecx.images-amazon.com\/images\/I\/418i8uQYB2L._SL110_.jpg","Height":{"_":"110","Units":"pixels"},"Width":{"_":"73","Units":"pixels"}},"MediumImage":{"URL":"http:\/\/ecx.images-amazon.com\/images\/I\/418i8uQYB2L._SL160_.jpg","Height":{"_":"160","Units":"pixels"},"Width":{"_":"107","Units":"pixels"}},"LargeImage":{"URL":"http:\/\/ecx.images-amazon.com\/images\/I\/418i8uQYB2L.jpg","Height":{"_":"500","Units":"pixels"},"Width":{"_":"333","Units":"pixels"}},"Category":"primary"}},"ItemAttributes":{"Author":["Elliotte Rusty Harold","W. Scott Means"],"Binding":"Paperback","CatalogNumberList":{"CatalogNumberListElement":"AL11998603"},"EAN":"9780596007645","EANList":
{"EANListElement":"9780596007645"},"Edition":"Third Edition","ISBN":"0596007647","ItemDimensions":{"Height":{"_":"906","Units":"hundredths-inches"},"Length":{"_":"610","Units":"hundredths-inches"},"Weight":{"_":"216","Units":"hundredths-pounds"},"Width":{"_":"150","Units":"hundredths-inches"}},"Label":"O'Reilly Media","Languages":{"Language":
[{"Name":"English","Type":"Unknown"},{"Name":"English","Type":"Original Language"},{"Name":"English","Type":"Published"}]},"ListPrice":
{"Amount":4499,"CurrencyCode":"USD","FormattedPrice":"$44.99"},"Manufacturer":"O'Reilly Media","NumberOfItems":1,"NumberOfPages":600,"PackageDimensions":{"Height":
{"_":"165","Units":"hundredths-inches"},"Length":{"_":"898","Units":"hundredths-inches"},"Weight":{"_":"190","Units":"hundredths-pounds"},"Width":{"_":"606","Units":"hundredths-inches"}},"PackageQuantity":1,"ProductGroup":"Book","ProductTypeName":"ABIS_BOOK","PublicationDate":"2004-09","Publisher":"O'Reilly Media","SKU":"G9780596007645","Studio":"O'Reilly Media","Title":"XML in a Nutshell, Third Edition"},"OfferSummary":{"LowestNewPrice":{"Amount":1953,"CurrencyCode":"USD","FormattedPrice":"$19.53"},"LowestUsedPrice":
{"Amount":355,"CurrencyCode":"USD","FormattedPrice":"$3.55"},"TotalNew":"41","TotalUsed":"59","TotalCollectible":"0","TotalRefurbished":"0"},"EditorialReviews":{"EditorialReview":
{"Source":"Product Description","Content":"

If you're a developer working with XML, you know there's a lot to know about XML, and the XML space is evolving almost moment by moment. But you don't need to commit every XML syntax, API, or XSLT transformation to memory; you only need to know where to find it. And if it's a detail that has to do with XML or its companion standards, you'll find it--clear, concise, useful, and well-organized--in the updated third edition of *XML in a Nutshell<\/i>.<\/p>*

With XML in a Nutshell<\/i> beside your keyboard, you'll be able to:<\/p>

- *Quick-reference syntax rules and usage examples for the core XML technologies, including XML, DTDs, Xpath, XSLT, SAX, and DOM <\/li>*
- *Develop an understanding of well-formed XML, DTDs, namespaces, Unicode, and W3C XML Schema <\/li>*
- *Gain a working knowledge of key technologies used for narrative XML documents such as web pages, books, and articles technologies like XSLT, Xpath, Xlink, Xpointer, CSS, and XSL-FO <\/li>*
- *Build data-intensive XML applications<\/li>*
- *Understand the tools and APIs necessary to build data-intensive XML applications and process XML documents, including the event-based Simple API for XML (SAX2) and the tree-oriented Document Object Model (DOM)<\/li><\/ul> This powerful new edition is the comprehensive XML reference. Serious users of XML will find coverage on just about everything they need, from fundamental syntax rules, to details of DTD and XML Schema creation, to XSLT transformations, to APIs used for processing XML documents. XML in a Nutshell<\/i> also covers XML 1.1, as well as updates to SAX2 and DOM Level 3 coverage. If you need explanation of how a technology works, or just need to quickly find the precise syntax for a particular piece, XML in a Nutshell<\/i> puts the information at your fingertips.*

  Simply put, XML in a Nutshell<\/i> is the critical, must-have reference for any XML developer.<\/p><\/div>","IsLinkSuppressed":false}}}}}