Assignment Jumpstart

Radu Nicolescu
Department of Computer Science
University of Auckland

27 July 2015

1. Data Sources
2. XML and XPATH
3. JSON
4. JSON/ODATA
5. OData and Browsers
6. API Hints
7. Code samples

Data Sources

- XML – NO XML schemas, NO namespaces
- XPATH
- JSON
- REST – basics
- ODATA – basic queries

XML

XML – NO XML schemas, NO namespaces, ...

XML Tutorial:

`http://www.w3schools.com/xml/default.asp`

Study these chapters: XML HOME, XML Introduction, XML How to use, XML Tree, XML Syntax, XML Elements, XML Attributes, XML Encoding.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<note date="2008-01-10">
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
</note>
```

XPATH

XPATH Tutorial:

`http://www.w3schools.com/xpath/default.asp`

Study these chapters: XPath HOME, XPath Nodes, XPath Syntax, XPath Axes, XPath Operators, XPath Examples.

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
<book @price="30.00">
 <title>XML</title> <author>Austen Powers</author>
</book>
<book @price="39.95">
 <title>XPATH</title> <author>Austen Powers</author>
</book>
</bookstore>
```

```
/bookstore/book[@price>=35]/price
```

JSON

JSON Tutorial:

`http://www.w3schools.com/json/default.asp`

Study these chapters: JSON HOME, JSON Intro, JSON Syntax.

```
{"employees":[
  {"firstName":"John", "lastName":"Doe"},
  {"firstName":"Anna", "lastName":"Smith"},
  {"firstName":"Peter", "lastName":"Jones"}
]}
```

- one outer object { }
- with one field named employees
- whose value is an array [ ] of three objects { }
- each array element has two fields, named firstName and lastName ...

JSON

Possible XML mapping:

```
{"employees":[
  {"firstName":"John", "lastName":"Doe"},
  {"firstName":"Anna", "lastName":"Smith"},
  {"firstName":"Peter", "lastName":"Jones"}
]}
```

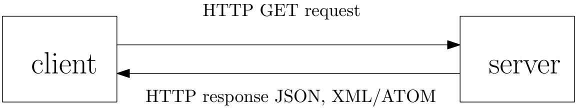
```
<employees>
  <employee>
    <firstName>John</firstName>
    <lastName>Doe</lastName>
  </employee>
   ...
</employees>
```

JSON — assignment file

```
{ "value":[
      {"OrderID":10000, "CustomerID":"VOID", ...},
      {"OrderID":10249, "CustomerID":"TOMSP", ...},
      ...
      {"OrderID":11077, "CustomerID":"RATTC", ...}
   ]}
```

- outer object { }
- with one field
  - value : array with 831 elements !

REST and ODATA



REST Introductions:

`http://www.drdobbs.com/web-development/restful-web-services-a-tutorial/240169069?pgno=1`

`http://www.infoq.com/articles/rest-introduction`

ODATA Introductions:

`http://www.odata.org/` : only steps 1-3, HTTP requests

`https://msdn.microsoft.com/en-us/library/ff478141.aspx` : focus on Query Navigation, Query Options, Filter Expressions

JSON — ODATA without truncation

```
http://services.odata.org/
    Northwind/Northwind.svc/Orders()?$orderby=OrderID
    $select=OrderID,CustomerID&$format=json
```

```
{
  "odata.metadata":"http://services.odata.org/...",
  "value":[
    {"OrderID":10248, "CustomerID":"VINET"},
    {"OrderID":10249, "CustomerID":"TOMSP"},
    ...
    {"OrderID":11077, "CustomerID":"RATTC"}
  ]}
```

- outer object { }
- with two fields
    - odata.metadata
    - value : array with 830 elements !

JSON — ODATA with truncation

```
{
  "odata.metadata":"http://services.odata.org/...",
  "value":[
    {"OrderID":10248, "CustomerID":"VINET"},
    {"OrderID":10249, "CustomerID":"TOMSP"},
    ...
    {"OrderID":10447,"CustomerID":"RICAR"}
  ],
  "odata.nextLink":"Orders?$select=OrderID,CustomerID&
                              $skiptoken=10447"
}
```

- outer object { }
- with three fields
  - odata.metadata
  - value : array with 200 elements only !
  - odata.nextLink : link to next chunk as relative URL

OData and Browsers

- ODATA is created for machine/application consumption, not for human readers
- By default, browsers are not prepared to display ODATA results and must be “helped” if we wish so
- IE
  - XML/ATOM: switch off Internet Options / Content / Feeds / reading view ☺
  - JSON: registry change ☹
- FF
  - JSON: JsonViewer add-on ☺
  - JSON and XML/ATOM: REST Client add-on ☺☺
- Fiddler
  - can be used standalone
  - or to trace browser activity

API Hints

- Suggestion to read more from the MSDN site...
- reading a textfile as a string: File.ReadAllText
- loading an XML doc from a string: XDocument.Parse
- loading an XML doc from a textfile: File.ReadAllText+XDocument.Parse *or* XDocument.Load
- saving an XML document to a text file: XDocument.Save
- fundamental .NET types for XML: XDocument, XElement, XAttribute; also (historically) XmlDocument
- fundamental XPATH methods XPathEvaluate, XPathSelectElements

API Hints

- fundamental LINQ methods: OrderBy, ThenBy, Select, SelectMany, Join, GroupJoin
- compare strings by ordinal char values: StringComparer.Ordinal
- to download from the Web as a string: Uri, WebClient, WebClient.DownloadString
- convert from JSON to XML: JsonConvert.DeserializeXmlNode (from JSON.NET library)
- for this conversion, suggestion to look at `http://www.newtonsoft.com/json/help/html/Introduction.htm`

XML file and XPATH selection

Consider the following XML document, in the file MyOrders.xml

```
<?xml version="1.0" encoding="utf-8"?>
<Orders>
  <Order OrderID="4010" CID="pdel">cuda card</Order>
  <Order OrderID="1020" CID="rnic">optical mouse</Order>
  <Order OrderID="1010" CID="rnic">flash memory</Order>
  <Order OrderID="2030" CID="sman">digital camera</Order>
  <Order OrderID="2020" CID="sman">pocket pc</Order>
  <Order OrderID="2010" CID="sman">iphone</Order>
</Orders>
```

Problem: select all orders with attribute OrderID > 2000

XML file and XPATH selection

Solution: load an XML document from the given file and select the required sequence of elements using XPATH

```
var xdoc = XDocument.Parse(
                File.ReadAllText("MyOrders.xml"));
var xpath = xdoc.XPathSelectElements(
                "Orders/Order[@OrderID > 2000]");
xpath.Dump("xpath");
```

Result:

```
<Order OrderID="4010" CID="pdel">cuda card</Order>
<Order OrderID="2030" CID="sman">digital camera</Order>
<Order OrderID="2020" CID="sman">pocket pc</Order>
<Order OrderID="2010" CID="sman">iphone</Order>
```

ODATA JSON

Problem: download the top three orders from our ODATA site, in the OrderID order, and select the OrderID and CustomerID fields

Solution:

```
var u = new Uri(
  "http://services.odata.org/Northwind/Northwind.svc/
  Orders?$orderby=OrderID&$top=3&
  $select=OrderID,CustomerID&$format=json");
var w = new System.Net.WebClient();
var json = w.DownloadString(u);
json.Dump("json");
```

Result:

```
{ "odata.metadata":"http://services.odata.org/...",
  "value":[
    {"OrderID":10248,"CustomerID":"VINET"},
    {"OrderID":10249,"CustomerID":"TOMSP"},
    {"OrderID":10250,"CustomerID":"HANAR"}]}
```