Random Thoughts!!!

Monday, May 8, 2017

XML parsing in AS400 (IBM-i) using XML-INTO

Hello Friends,

If you are new in working with XML in AS400 or just starting to do some XML parsing, this article will be useful to you.

XML-INTO is the op-code used to parse the XML document and retrieve the values directly into the variables in RPGLE.

Previously I saw many examples using DOM parser but it is complex for any beginners. But IBM introduced XML-INTO op-code and made the XML parsing much easier. Let us directly jump into syntax and examples.

Syntax:

xml-into <VariableName> %xml(<filename> : options)

There are couple of options which are considered while parsing the xml. We will see one by one.

Mostly the <variablename> will be a data structure which matches with the xml structure.

Simple XML:

```
 Browse : /home/devyus/input.xml
 Record :      1    of       6 by  18
 Control :

....+....1....+....2....+....3....+....4....+.
 ************Beginning of data**************
<?xml version="1.0" encoding="utf-8"?>
<Customer>
  <CustName>Yusuf</CustName>
  <Status>Active</Status>
  <CustId>123</CustId>
</Customer>
 ************End of Data*********************
```

RPGLE:

```
0008.00 D
0009.00 D FileName        S             50A   Inz
0010.00 D Options         S             50A   Inz
0011.00 D
0012.00 D Customer        DS                  Qualified
0013.00 D  CustName                     20
0014.00 D  Status                       10
0015.00 D  CustId                       10S 0
0016.00 D
0017.00  *-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
0018.00 C
0019.00  /Free
0020.00
0021.00       options = 'case=any doc=file' ;
0022.00
0023.00       FileName = '/Home/Devyus/Input.xml';
0024.00
0025.00       xml-into Customer
0026.00          %xml(%Trim(FileName):options);
0027.00
0028.00       Dsply  Customer;
0029.00
0030.00       Return;
0031.00  /End-Free
```

Output:

About Me

mohammed yusuf m

G+ Follow 16

View my complete profile

Blog Archive

- ▼ 2017 (9)
  - ▼ May (2)
    - XML parsing in AS400 (IBM-i) using XML-INTO
    - AJAX & JSON with CGI Programming in IBM-I
  - ► April (5)
  - ► March (2)

Total Pageviews

4686

```
> EVAL Customer
  CUSTOMER.CUSTNAME = 'Yusuf
  CUSTOMER.STATUS = 'Active    '
  CUSTOMER.CUSTID = 0000000123.
```

Options in xml-into:

Case=any

In general XML tag names should match with RPG variables with case sensitive. But using case=any option we can eliminate it.

Doc=file

This tells the op-code that xml data source is coming from a file document. In this case, the first parameter of %XML function will be file name.

Allowmissing=yes

Sometimes if we are having missing fields in XML document

Allowextra=yes

Sometimes if we get extra tag in the XML (not defined in DS)

In other words, if u expect all the DS fields are mandatory in your XML you can give allowmissing=no and handle the exception and allowextra=no will not accept any extra tag to come in your xml.

In most cases, we use both as 'yes' and handle the exception in program.

Path="string"

Using path option you can directly read any particular tag value. We will see it deeper in coming examples.

A complex XML:

Let us look at below example which has nested tags, attribute value and also repeating a particular tag number of times.

```
<Customer id="123">          <-- We have attribute value
  <Name>Yusuf</Name>
  <Status>Active</Status>
  <ContactInfo>
    <WorkPhone>123123123</WorkPhone>
    <MobilePhone>321321321</MobilePhone>
  </ContactInfo>
  <Points>25</Points>
  <Order>
    <OrderNumber>H12345</OrderNumber>
    <OrderDetail>
      <Item>                 <-- <Item> Tag can repeat
        <ItemName>Item1</ItemName>
        <Quantity>2</Quantity>
        <DepartmentNumber>1</DepartmentNumber>
      </Item>
      <Item>
        <ItemName>Item2</ItemName>
        <Quantity>2</Quantity>
        <DepartmentNumber>2</DepartmentNumber>
      </Item>
      <Item>
        <ItemName>Item3</ItemName>
        <Quantity>2</Quantity>
        <DepartmentNumber>3</DepartmentNumber>
      </Item>
    </OrderDetail>
  </Order>
</Customer>
```

Here we need to know how to define a nested data structure and how to find number of times a tag being repeated. Fortunately, xml-into offers an easy way to do that.

And trust me we can just parse all the values with single statement ☺

How to form nested Data Structure:

```
D
D Customer        DS                  Qualified
D  ID                              5
D  Name                           20
D  Status                         10
D  ContactInfo                    LikeDS(CtInfo_T)
D  Points                       5S 0
D  Order                          LikeDS(Order_T)
D
D CtInfo_T        DS                  Template
D  WorkPhone                      10
D  MobilePhone                    10
D
D Order_T         DS                  Qualified
D  OrderNumber                    10
D  OrderDetail                    LikeDS(OrdDet_T)
D
D OrdDet_T        DS                  Qualified
D  Item                           LikeDS(Item_T) Dim(10)
D  Count_Item                   5i 0
D
D
D Item_T          DS                  Template
D  ItemName                       15A
D  DepartmentNumber...
D                                 2S 0
D  Quantity                       5S 0
D
```

If you look closely, we have split the group of tags into separate DS and nested into main Customer DS using LikeDS keyword. The reason for CtInfo_T & Item_T is declared as Template but Order_T & OrdDet_T are declared as Qualified because, if we are going to have any nested DS then the parent DS should be declared as Qualified.

Take a note on DepartmentNumber. This exceeds normal variable length of 15 char while defining in RPGLE. So if we have variable length more than 15 char then we have to write the full name followed by three dots. Thus system will look for its declaration in next line. (in our case 2S 0)

Count_Item:

Do notice this variable Count_Item is an extra field apart from XML fields. When we need to parse multiple occurrence tags (in our case <item> tag) then we have to define a variable next to repeated tag so that system will automatically give us the number of repetition in this variable.

RPGLE code:

```
 /Free

      options = 'doc=file case=any ' +
                'allowmissing=yes allowextra=yes ' +
                'countprefix=count_';

      FileName = '/Home/Devyus/Input.xml';

      xml-into customer
         %xml(%Trim(FileName):options);

      Return;
 /End-Free
C
```

Please note on the countprefix option. We are saying the system to return the count of repetition under the variable has prefix “count_”

i.e. for item tag the count variable should be “count_item”

if we change “countprefix=cnt_” then in our DS, the variable should be “cnt_item”

here we go, we got all the values, and also the count_item has number of repetition as “3”.

```
> EVAL customer
  CUSTOMER.ID = '123   '
  CUSTOMER.NAME = 'Yusuf                '
  CUSTOMER.STATUS = 'Active     '
  CUSTOMER.CONTACTINFO.WORKPHONE = '123123123 '
  CUSTOMER.CONTACTINFO.MOBILEPHONE = '321321321 '
  CUSTOMER.POINTS = 00025.
  CUSTOMER.ORDER.ORDERNUMBER = 'H12345     '
  CUSTOMER.ORDER.ORDERDETAIL.ITEM.ITEMNAME(1) = 'Item1
  CUSTOMER.ORDER.ORDERDETAIL.ITEM.DEPARTMENTNUMBER(1) = 01.
  CUSTOMER.ORDER.ORDERDETAIL.ITEM.QUANTITY(1) = 00002.
  CUSTOMER.ORDER.ORDERDETAIL.ITEM.ITEMNAME(2) = 'Item2
  CUSTOMER.ORDER.ORDERDETAIL.ITEM.DEPARTMENTNUMBER(2) = 02.
  CUSTOMER.ORDER.ORDERDETAIL.ITEM.QUANTITY(2) = 00004.
  CUSTOMER.ORDER.ORDERDETAIL.ITEM.ITEMNAME(3) = 'Item3          '
  CUSTOMER.ORDER.ORDERDETAIL.ITEM.DEPARTMENTNUMBER(3) = 03.
  CUSTOMER.ORDER.ORDERDETAIL.ITEM.QUANTITY(3) = 00006.
  CUSTOMER.ORDER.ORDERDETAIL.ITEM.ITEMNAME(4) = '                '
  CUSTOMER.ORDER.ORDERDETAIL.ITEM.DEPARTMENTNUMBER(4) =    .
  CUSTOMER.ORDER.ORDERDETAIL.ITEM.QUANTITY(4) =       .
  CUSTOMER.ORDER.ORDERDETAIL.ITEM.ITEMNAME(5) = '                '
  CUSTOMER.ORDER.ORDERDETAIL.ITEM.DEPARTMENTNUMBER(5) =    .
  CUSTOMER.ORDER.ORDERDETAIL.ITEM.QUANTITY(5) =       .
  CUSTOMER.ORDER.ORDERDETAIL.COUNT_ITEM = 3
```

With this in hand, we can very well do our program logic and proceed further.

One last example using path option:

Using path we can directly read any segment of xml. For example, I want to retrieve OrderNumber alone from the XML. Then,

```
0052.00
0053.00        options = 'doc=file case=any ' +
0054.00                  'path=Customer/ContactInfo/OrderNumber';
0055.00
0056.00        xml-into OrdNum
0057.00           %xml(%Trim(FileName):options);
0058.00
```

Do note that if we are fetching value of a single tag directly then variable name to receive the value may not need to be the same xml tag name. (Here OrdNum is used).

Hope this post gives you enough idea to get start with xml-into method of parsing. It is really very useful as you can see it requires very minimal amount of coding. I will come with another topic soon. Until then…

Have fun..!!! Happy coding…!!!

Posted by mohammed yusuf m at 3:13 AM

Labels: XML in AS400, XML parsing in AS400, XML-INTO

No comments:

Post a Comment