文章
姚 鑫 · 六月 16, 2021 阅读大约需 7 分钟

第九章 将XML导入到对象中

第九章 将XML导入到对象中

本章介绍如何使用%XML.Reader将XML文档导入到 IRIS对象中。

注意：使用的任何XML文档的XML声明都应该指明该文档的字符编码，并且文档应该按照声明的方式进行编码。如果未声明字符编码， IRIS将使用前面的“ 输入和输出的字符编码 ”中描述的默认值。如果这些默认值不正确，请修改XML声明，使其指定实际使用的字符集。

还可以使用%XML.Reader读取任意XML文档并返回DOM(文档对象模型)。

创建XML读取器概述

IRIS提供了一些工具，用于读取XML文档并创建与该文档的元素相对应的启用XML的
IRIS对象的一个或多个实例。基本要求如下：

- 该对象的类定义必须扩展%XML.Adaptor。除了少数例外，该对象引用的类还必须扩展%XML.Adaptor。

提示：如果相应的XML模式可用，可以使用它来生成类(以及任何支持的类)。

- 要导入XML文档，创建%XML.Reader的实例，然后调用该实例的方法。这些方法指定XML源文档，将XML元素与启用XML的类相关联，并将源中的元素读取到对象中。

%XML.Reader使用类中的%XML.Adaptor提供的方法执行以下操作：

- 它使用InterSystems IRIS SAX接口解析和验证传入的XML文档。验证可以包括DTD或XML架构验证。
- 它确定是否有任何启用了XML的对象与XML文档中包含的元素相关，并在读取文档时创建这些对象的内存中实例。

请注意，%XML.Reader创建的对象实例不存储在数据库中；它们是内存中的对象。如果要将对象存储在数据库中，则必须调用%Save()方法(对于持久对象)，或者将相关属性值复制到持久对象并保存它。应用程序还必须决定何时插入新数据和何时更新现有数据；%XML.Reader无法进行此区分。

下面的终端会话显示了一个简单的示例。在这里，我们将XML文件读入一个新对象，检查该对象，然后保存该对象：

```
/// w ##class(PHA.TEST.Xml).ReadXml()
ClassMethod ReadXml()
{
    Set reader = ##class(%XML.Reader).%New()
    Set file="E:\temp\samplePerson.xml"
    Set status = reader.OpenFile(file)
    if $$$ISERR(status) { do $System.Status.DisplayError(status) quit }
    Write status,!
```

```
    Do reader.Correlate("Person","Sample.Person")
    Do reader.Next(.object,.status)
    if $$$ISERR(status) { do $System.Status.DisplayError(status) quit }
    Write object.Name,!

    Do object.%Save()
    q ""
}
```

此示例使用以下示例XML文件：

```
<Person GroupID="90455">
  <Name>Worthington,Jeff R.</Name>
  <DOB>1976-11-03</DOB>
  <Address>
    <City>Elm City</City>
    <Zip>27820</Zip>
  </Address>
  <Doctors>
    <Doctor>
      <Name>Best,Nora A.</Name>
    </Doctor>
    <Doctor>
      <Name>Weaver,Dennis T.</Name>
    </Doctor>
  </Doctors>
</Person>
```

创建导入方法

总体方法结构

方法应按以下顺序执行以下部分或全部操作：

1. 创建%XML.Reader的实例。
2. 也可以指定此实例的Format属性，以指定要导入的文件的格式。

默认情况下，InterSystems
IRIS假定XML文件为文字格式。如果文件是SOAP编码格式，则必须指明这一点，以便可以正确读取该文件。

3. 可以选择设置此实例的其他属性。
4. 请使用%XML.Reader的以下方法之一

- OpenFile() -打开文件。
- OpenStream()-打开一个流。
- OpenString() -打开一个字符串。
- OpenURL() -打开一个URL。

在每种情况下，可以选择性地为该方法指定第二个参数，以覆盖Format属性的值。

5. 将这个文件中的一个或多个XML元素名与具有相应结构的支持InterSystems IRIS XML的类关联起来。
   有两种方法可以做到这一点:

- 使用Correlate()方法，它有以下签名:

```
method Correlate(element As %String,
                 class As %String,
                 namespace As %String)
```

其中element是XML元素名，class是InterSystems IRIS类名(带包)，namespace是可选的名称空间URI。

如果使用namespace参数，则匹配仅限于指定命名空间中的指定元素名。
如果将命名空间参数指定为""，则与Next()方法中给出的默认命名空间相匹配。
如果不使用namespace参数，则只使用元素名进行匹配。

提示:可以反复调用Correlate()方法来关联多个元素。

- 使用CorrelateRoot()方法，它有以下签名:

```
method CorrelateRoot(class As %String)
```

其中class是InterSystems IRIS类名(带包)。此方法指定XML文档的根元素与指定的类相关。

6. 按如下方式实例化类实例：

如果使用Correlate()，则遍历文件中的相关元素，一次循环一个元素。在循环中，使用Next()方法，该方法具有以下签名：

```
method Next(ByRef oref As %ObjectHandle,
            ByRef sc As %Status,
            namespace As %String = "") as %Integer
```

其中OREF是该方法创建的对象，sc是状态，Namespace是文件的默认名称空间。

- 如果使用CorrelateRoot()，请调用next()方法一次，这会导致实例化相关类。

Next()方法在到达文件末尾时返回0。如果在此之后再次调用next()，则将从文件顶部开始再次循环遍历文件中的对象。(指定的关联仍然有效。)

错误检查

上一节提到的大多数方法都返回状态。应该在每个步骤之后检查状态，并在适当的情况下退出。

基本导入示例

名为test.xml的以下XML文件：

```
<Root>
   <Person>
      <Name>?</Name>
   </Person>
   <Person>
```

```
      <Name>?</Name>
   </Person>
</Root>
```

我们首先定义一个启用XML的类MyApp.Person，它是Person的对象表示：

```
Class MyApp.Person Extends (%Persistent, %XML.Adaptor)
{

Parameter XMLNAME = "Person";

Property Name As %String;

Storage Default
{
<Data name="PersonDefaultData">
<Value name="1">
<Value>%%CLASSNAME</Value>
</Value>
<Value name="2">
<Value>Name</Value>
</Value>
</Data>
<DataLocation>^MyApp.PersonD</DataLocation>
<DefaultData>PersonDefaultData</DefaultData>
<IdLocation>^MyApp.PersonD</IdLocation>
<IndexLocation>^MyApp.PersonI</IndexLocation>
<StreamLocation>^MyApp.PersonS</StreamLocation>
<Type>%Library.CacheStorage</Type>
}

}
```

要将此文件导入到MyAppPerson类的实例中，我们可以编写以下方法：

```
/// w ##class(PHA.TEST.Xml).ImportXml()
ClassMethod ImportXml()
{
    // ??%XML.Reader???
    Set reader = ##class(%XML.Reader).%New()

    // ??????
    Set status = reader.OpenFile("E:\temp\testPerson.xml")
    If $$$ISERR(status) {do $System.Status.DisplayError(status)}

    // ????XML??????
    Do reader.Correlate("Person","MyApp.Person")

    // ?XML??????
    While (reader.Next(.object,.status)) {
        Write object.Name,!
    }
```

```
    // ??????????????????
    If $$$ISERR(status) {do $System.Status.DisplayError(status)}
    q ""
}


DHC-APP>w ##class(PHA.TEST.Xml).ImportXml()
?
?
```

此方法执行几个任务：

- 它使用InterSystems IRIS SAX接口解析输入文件。这包括根据文档的DTD或架构(如果指定)验证文档。
- Correlate()方法将类MyApp关联起来。
  MyPerson与XML元素<Person>;
  <Person>中的每个子元素都成为MyPerson的一个属性。
- 它从输入文件中读取每个<Person>元素，直到没有剩余元素。
- 最后，如果循环因错误而终止，则该错误将显示在当前输出设备上。

如上所述，此示例不将对象存储到数据库。因为MyPerson是持久对象，所以可以通过在While循环中添加以下行来完成此操作：

```
/// w ##class(PHA.TEST.Xml).ImportXml()
ClassMethod ImportXml()
{
    // ??%XML.Reader???
    Set reader = ##class(%XML.Reader).%New()

    // ??????
    Set status = reader.OpenFile("E:\temp\testPerson.xml")
    If $$$ISERR(status) {do $System.Status.DisplayError(status)}

    // ????XML??????
    Do reader.Correlate("Person","MyApp.Person")

    // ?XML??????
    While (reader.Next(.object,.status)) {
        Write object.Name,!
        Set savestatus = object.%Save()
        If $$$ISERR(savestatus) {do $System.Status.DisplayError(savestatus)}
    }

    // ??????????????????
    If $$$ISERR(status) {do $System.Status.DisplayError(status)}


    q ""
}
```

通过HTTPS URL访问文档

对于OpenURL()方法，如果文档位于需要SSL/TLS的URL，请执行以下操作：

1. 使用管理门户创建包含所需连接详细信息的SSL/TLS配置。这是一次性的步骤。
2. 使用%XML.Reader时，请设置读取器实例的SSLConfiguration属性。对于该值，请指定在上一步中创建的SSL/TLS配置的名称。

或者，当使用%XML.Reader，还可以执行以下操作：

1. 创建%Net.HttpRequest实例。
2. 将该实例的SSLConfiguration属性设置为等于管理门户中创建的SSL/TLS配置的配置名称。
3. 使用%Net.HttpRequest的实例作为OpenURL()的第三个参数。

例如：

```
Class YX.Config Extends (%Persistent, %XML.Adaptor)
{

Parameter XMLNAME = "update";

Property version As %String;

Property name As %String;

Property url As %String;
}


/// ??http?xml???????
/// w ##class(PHA.TEST.Xml).ReadXmlHttp("http://192.168.10.3/dthealth/web/csp/version
.xml")
ClassMethod ReadXmlHttp(url)
{
    set reader = ##class(%XML.Reader).%New()
    set request = ##class(%Net.HttpRequest).%New()
    set request.SSLConfiguration="yx"
    set status = reader.OpenURL(url,,request)
    If $$$ISERR(status) {do $System.Status.DisplayError(status)}

    // ????XML??????
    Do reader.Correlate("update","YX.Config")

    While (reader.Next(.object,.status)) {
        Write object.version,!
        Write object.name,!
        Write object.url,!
    }
    q ""
}


DHC-APP>w ##class(PHA.TEST.Xml).ReadXmlHttp("http://192.168.10.3/dthealth/web/csp/ver
sion.xml")
27
Herb
http://192.168.31.124/dthealth/web/csp/Herb.apk
```

在服务器需要身份验证时访问文档

如果服务器需要身份验证，请创建%Net.HttpRequest的实例，并设置该实例的用户名和密码属性。还可以如上所述使用SSL(因此还要设置SSLConfiguration属性)。然后使用%Net.HttpRequest的实例作为OpenURL()的第三个参数，如上例所示。

#Caché #InterSystems IRIS

源URL:
https://cn.community.intersystems.com/post/%E7%AC%AC%E4%B9%9D%E7%AB%A0-%E5%B0%86xml%E5%AF%BC%E5%85%A5%E5%88%B0%E5%AF%B9%E8%B1%A1%E4%B8%AD