Article
Yuri Marx · Jan 16, 2021 3m read

Creating and listing XData definitions

Hi InterSystems Community!

The ObjectScript language of InterSystems IRIS has the ability to extend classes using a very interesting feature called XData.

It is a section in your class that can be used to create custom definitions to be used within the class itself and also externally.

To create one or more XData definitions for your class is very easy, see the example:

```
Class dc.Sample.Person Extends (%Persistent, %JSON.Adaptor, %Populate)
{
 Property Name As %VarString;
 Property Title As %String;
 Property Company As %String;
 Property Phone As %VarString;
 Property DOB As %Date(MAXVAL = "$piece($horolog, "","", 1)");
 /// Index for property DOB
Index DOBIndex On DOB;
 ClassMethod AddTestData(amount As %Integer = 10)
{
  d ..Populate(amount)
}
 /// Documentation for Person
XData PersonDocHtml [ MimeType = text/html ]
{
<h1>This is the Person class</h1>
}
 XData PersonDocMarkdown [ MimeType = text/markdown ]
{
<h1>This is the Person class</h1>
}
```

Note that after defining the methods, just add one or more XData sections with three sections: XData NomeSecaoXData [MimeType = TypeOfMimeType]. The content is then placed between {}.

All XData elements are stored in the% Dictionary.XDataDefinition persistent class. This means that it is possible to retrieve the definitions using SQL language, internally or externally, see the example:

```
 Class dc.mkdocs.Generator
```

```
{
  ClassMethod Generate()
{
   Set qry =
"SELECT parent, Name, Description FROM %Dictionary.XDataDefinition WHERE
 MimeType IN ('text/markdown','text/html')"
   Set stm = ##class(%SQL.Statement).%New()
   Set qStatus = stm.%Prepare(qry)
   If qStatus'=1 {Write "%Prepare failed:" Do $System.Status.
DisplayError(qStatus) Quit}
   Set rset = stm.%Execute()
   While rset.%Next() {
      Write "Row count ",rset.%ROWCOUNT,!
      Write rset.parent
      Write ": ",rset.Description,!
      Write ..GetXDataContent(rset.parent,rset.Name),!!
   }
   Write !,"Total row count=",rset.%ROWCOUNT
}
  ClassMethod GetXDataContent(className, xdataName) As %String
{
   set content = ""
   for i=1:1:$$$comMemberKeyGet(className,
$$$cCLASSxdata,xdataName,$$$cXDATAdata) {
      set content = content_$$$comMemberArrayGet(className,
$$$cCLASSxdata,xdataName,$$$cXDATAdata,i)
   }
   quit content
}
  }
```

In this example, all XData elements with Mime Type markdown and html are retrieved and then have the name of the class where the XData is located and the description of the XData printed. If you want to retrieve the content, see GetXDataContent (thanks @Eduard Lebedyuk).

It is a very interesting feature, as we can catalog the documentation of all classes of an application and have easy access to them. Fantastic!

#ObjectScript #InterSystems IRIS