Artículo
Eduardo Anglada · 25 mayo, 2021 Lectura de 3 min

Cómo crear y registrar definiciones XData

¡Hola Comunidad!

ObjectScript, el lenguaje de InterSystems IRIS, tiene la capacidad de extender clases utilizando una funcionalidad muy interesante llamada XData.

Es una sección en tu clase que puede ser usada para crear definiciones personalizadas, para utilizarlas dentro de la clase misma o externamente.

Crear una o más definiciones XData para tu clase es muy sencillo, mira el ejemplo:

```
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>
}
```

Ten en cuenta que tras definir los métodos, solo tienes que añadir una o más secciones XData con tres subsecciones: XData NomeSecaoXData [MimeType = TypeOfMimeType]. El contenido se pone entre corchetes.

Todos los elementos XData se almacenan en la clase persistente % Dictionary.XDataDefinition. Esto significa que es posible recuperar las definiciones usando lenguaje SQL, internamente o externamente. Mira el ejemplo:

```
 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
}
  }
```

En este ejemplo, todos los elementos XData con *Mime Type* markdown y html son recuperados y se imprime tanto el nombre de la clase en la que se localiza el elemento XData como su descripción. Si quieres recuperar el contenido del elemento XData consulta GetXDataContent (gracias @Eduard Lebedyuk).

Es una funcionalidad muy interesante, ya que podemos catalogar la documentación de todas las clases de una aplicación y tener un fácil acceso a ellas. ¡Genial!

#ObjectScript #InterSystems IRIS