Artículo
Heloisa Paiva · 6 sep, 2022 Lectura de 4 min

Interoperabilidad - Paso a paso: creando y conectando business hosts

Recientemente empecé a estudiar interoperabilidad y la documentación oficial fue muy útil para comprender la operación de los business hosts, pero aún me costó lograr hacerlo con mis manos. Mis compañeros de trabajo me ayudaron hasta que logré crear una Demo de un sistema y aprender practicando. Por eso, quise escribir acá para pasar adelante la ayuda que tuve.

Introducción

Antes de todo, no podemos olvidar los conceptos:

- Interoperabilidad - el significado no es tan complejo como su pronunciación - interoperabilidad es el nombre para la " magia " que trae y entrega todo tipo de información de un sistema al otro.
- Business hosts - si la interoperabilidad es la magia, los business hosts son los sombreros de copa del mago - hay los business services, que reconocen y reciben información, y las envían como mensajes a los business processes o business operations. Los business operations hacen las operaciones deseadas (como sugiere el nombre) y entregan el mensaje. Los business processes controlan el flujo de los mensajes: definen adonde va el mensaje y cómo es pasado adelante.

- Adaptadores - los adaptadores son clases que podemos usar para reconocer y manipular todo tipo de información que necesitamos tratar. En la práctica, usamos parámetros y propiedades y accedemos a tus métodos y propiedades.

Preparándose para crear la producción

Es más fácil empezar simple - vamos usar los services y operations primero - tenemos un service que recibe un tipo de mensaje que es fácilmente reconocido por la única operation con lo que estamos trabajando.

El desarrollo es más fácil cuándo el propósito de la producción y sus partes están muy claras. Si quieres, dibujar un diagrama o escribir los pasos que quieres lograr con el código puede ayudarte.

Por ejemplo:

Empezá preguntándote " ¿qué tengo que hacer? " - en mi Demo, necesité manipular una tabla SQL - yo doy informaciones como un Título y un Autor de un libro y inserto en la tabla.

" Entonces,¿qué necesito que la producción haga? " - tiene que recibir el mensaje conteniendo el Título y

el Autor y hacer un SQL INSERT

" Ok. ¿Cómo lo puedo lograr? " - el Business Service (BS) va a recibir el Título y el Autor y pasarlos al Business Operation (BO). El BO hace el código SQL.

" Ahora que tengo los caminos de la información, quiero entender el mensaje. ¿Qué es?" - hay muchas maneras de enviar los datos. Puedo usar un archivo, una aplicación REST, hasta un email. Vamos con el archivo para empezar simple. Mi BS va a recibir el archivo, leelo y enviar sus informaciónes hasta el BO, que va hacer la query.

Al trabajo

Puedes empezar por lo que te sientes más seguro.

- El Business Service

```
Class Demo.Books.BS.FileService Extends Ens.BusinessService
{

Parameter ADAPTER = "EnsLib.File.InboundAdapter";

Method OnProcessInput(pInput As %Stream.Object, Output pOutput As %RegisteredObject) As %Status
  {
      Set tSC = $$$OK

      Try
      {
          // Reads the first line of the recieved File
          Set tLine = pInput.ReadLine()

          // Sets the properties of the request based on the data recieved
          Set tRequest = ##class(Demo.Books.BO.Register.Request).%New()
          Set tRequest.Title = $PIECE(tLine, ",", 1)
          Set tRequest.Author = $PIECE(tLine, ",", 2)

          // Sends to operation
          Set tSC = ..SendRequestSync("Demo.Books.BO.Operation", tRequest, .tResponse)

          Throw:$$$ISERR(tSC)
      }
      Catch (tEx)
      {
          Set:'$$$ISERR(tSC) tSC = tEx.AsStatus()
      }

      Quit tSC
  }

}
```

Yo empecé con el BS. Ahora veo que necesito implementar la clase de Request, para que registres los datos, y el Operation a donde se va enviar.

- El Business Operation

Request

```
Class Demo.Books.BO.Register.Request Extends (Ens.Request, %XML.Adaptor)
{

Parameter RESPONSECLASSNAME = "Ens.Response";

Property Title As %String;

Property Author As %String;

}
```

La clase de Request es simple así, solo la voy usar para registrar datos. El %XML.Adaptor sirve para mostrar el Request en el Portal de Gestión para gestión de errores.

Operation

```
Class Demo.Books.BO.Operation Extends Ens.BusinessOperation
{

Property Adapter As EnsLib.SQL.OutboundAdapter;

Parameter ADAPTER = "EnsLib.SQL.OutboundAdapter";

Parameter INVOCATION = "Queue";

Method Register(pRequest As Demo.Books.BO.Register.Request, Output pResponse As Ens.Response) As %Status
 {
     // I could implement the method here, but to be more organized I created an Abstract Class for it.
     Quit ##class(Demo.Books.BO.Register.Method).Execute(##this, pRequest, .pResponse)
 }

// This map recognizes the type of message was sent in the request and executes the method specified.

XData MessageMap
 {
<MapItems>
 <MapItem MessageType = "Demo.Books.BO.Register.Request">
   <Method>Register</Method>
  </MapItem>
 </MapItems>
 }

}
```

El BO tiene un mapa de mensajes para dar las direcciónes para cada clase de mensaje que llega. Por ejemplo, si en el BS, en el método SendRequestSync, en el argumento Request, yo hubiera usado una otra clase, como “ Demo.Books.BO.SearchTable.Request ” , yo pudiera crear un otro MapItem con esa clase de mensaje en MessageType, con referencia a un método Search.

Method

```
Class Demo.Books.BO.Register.Method [ Abstract ]
{

ClassMethod Execute(pHost As Demo.Books.BO.Operation, pRequest As Demo.Books.BO.Register.Request, Output pResponse As Ens.Response) As %Status
 {
     Set tSC = $$$OK

     Try
     {
         Set tSC = pRequest.NewResponse(.pResponse)

         Throw:$$$ISERR(tSC)

         Set tSC = pHost.Adapter.ExecuteUpdate(.pRow, "INSERT books (title, author) VALUES (?,?)", pRequest.Title, pRequest.Author)

         Throw:$$$ISERR(tSC)
     }
     Catch (tEx)
     {
         Set:'$$$ISERR(tSC) tSC = tEx.AsStatus()
     }

     Quit tSC
 }

}
```

Acá, puedes implementar todo lo que tu Operation tiene que hacer. Es mejor implementar el método en otra clase, porque si hay que recompilar el BO, es posible que deba reiniciar la producción para que puedas trabajar con nuevos cambios.

Ajustes en el Portal de Gestión

Al fin, para que todo trabaje bien, siga los pasos:

Portal de Gestión > Interoperabilidad > Lista > Producciones > Nuevo

Así, el portal crea una clase con las informaciones de producción.

Entonces, puedes añadir un Service con la clase creada y establecer el camino del archivo (donde se van a poner los inputs) y un camino de trabajo.

También puedes añadir el Operation con la clase creada y especificar tus ajustes si es necesario.

Observaciones

- El Business Operation puede recibir una Synchronous Request (solicitud síncrona), ahí el Service solo responde cuando el BO ya retornó tu mensaje, porque la respuesta del Service necesita de la respuesta del BO, por ejemplo si hubiera hecho una operación pesquisa en la tabla. Acá, la producción solo hace una operación INSERT , el BS solo necesita enviar información y el BO insertalo; no necesita respuesta, entonces podría tener un Asynchronous Request (solicitud asincrónica)
- No está en el alcance de este artículo tratar de especificaciones como adaptadores File y SQL. Acá solo quiero mostrar una visión general del código y pasos para comprender mejor la práctica.
- Por favor, llamame para preguntas y ayuda!

#Innovatium #Interoperabilidad #InterSystems IRIS

URL de fuente:https://es.community.intersystems.com/post/interoperabilidad-paso-paso-creando-y-conectando-business-hosts