

Artículo

[Bernardo Linarez](#) · 22 nov, 2021 · Lectura de 2 min

Personalización de un Business Operation core en IRIS / Ensemble

Tal como sucede con otros tipos de Host, en ciertos casos de uso nos conviene personalizar el comportamiento de algún Business Operation (BO) de caja. En este caso, tomaré de ejemplo el siguiente `EnsLib.REST.GenericOperation`

En principio, el caso de uso era un simple Pass Through de una petición HTTP Rest, por lo que se decidió usar tanto el BS y el BO core de IRIS para reenviarlo. Sin embargo, el destino de esta petición, necesitaba algunos Header por defecto, que no proporcionaba el request original.

Para efectos del ejemplo, añadiré estos Header a través de un BO personalizado. En algunos casos de uso más complejos, lo mejor es realizarlos desde un Business Process (BP). Utilizaré la técnica de herencia provista en IRIS a través del lenguaje ObjectScript.

En el caso de los BO, se tiene que redefinir el método de clase `OnMessage` (a diferencia de los BS, que el método es el `OnProcessInput`), el cual se llama cuando llega un mensaje desde algún Host hacia el BO.

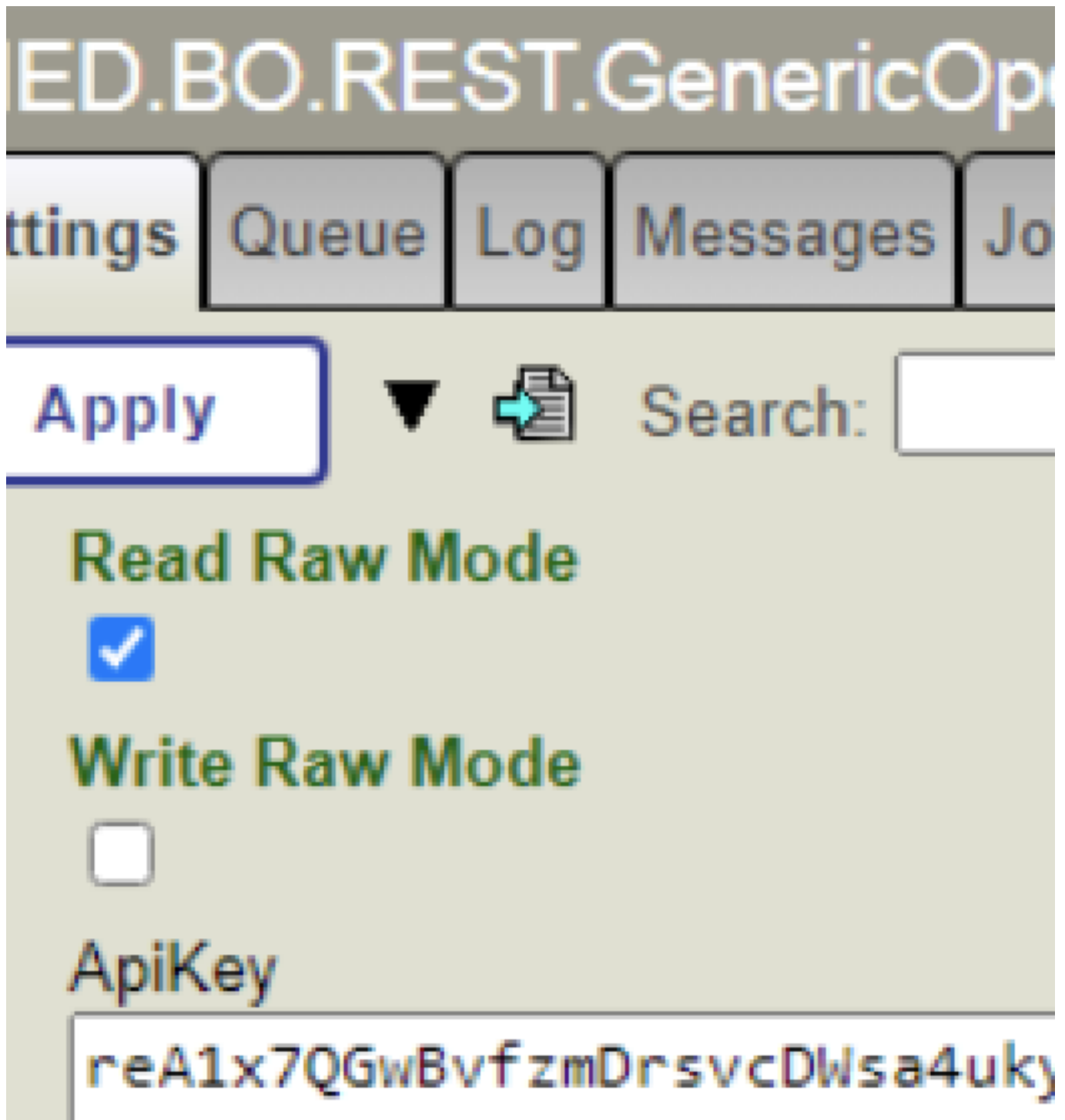
```
Class Test.BO.REST.GenericOperation Extends EnsLib.REST.GenericOperation
{
    Property ApiKey As %String(MAXLEN = "");

    Parameter SETTINGS = "ReadRawMode,WriteRawMode,ApiKey";

    Method OnMessage(pRequest As EnsLib.HTTP.GenericMessage, Output pResponse As EnsLib.
HTTP.GenericMessage) As %Status
    {
        Set tSC = $$$OK
        Try
        {
            Set tSC = pRequest.HTTPHeaders.SetAt(..ApiKey,"apikey")

            Set tSC = ##super(pRequest,.pResponse)
            Quit:$System.Status.IsError(tSC)
        }
    }
}
```

Notar que incluyo una propiedad al BO llamada "apikey", el cuál es necesario para invocar al endpoint REST. Este debe ser enviado vía HTTP Headers. Por otro lado, el parámetro `SETTINGS` incluye variables que existen en las clases padres; esto ocurre porque al definir el parámetro se reescribe toda la definición de las clases padres, entonces es la responsabilidad de la clase actual, incluirlos para que puedan ser administrados desde la UI del BO.



Luego de modificar el parámetro pRequest, recién se invoca a la lógica core original del método OnMessage, dado que es el que se encarga de resolver la petición HTTP Rest.

En este ejemplo, la lógica personalizada, se ejecuta previo a llamar a la lógica original. También es posible que primero se invoque a la lógica original y después la personalizada.

Espero les sea útil.

[#Ensemble](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

