

Announcement

[Neerav Verma](#) · Sep 17, 2018

PubSub - Publish & Subscribe Messages

Hello,

Fellow community members, here is an extendable / re-usable / generic pub-sub model.

An initial calling service reads a CSV file, parses through all records and transforms it to a generic JSON message.

The message is then transported via a pub-sub business process to the end-points Targets (business operations)

as configured for the subscribers for each Topic as demonstrated in the image below.

[Show Domains](#)[Show Subscribers](#)[Show Subscriptions](#)[Create New Subscription](#)

The following PubSub Subscriptions are currently defined. A subscription associates a Subscriber with

Filter:	<input type="text"/>	Page size:	20 ▼	Items found:	3
#	DomainName^	Topic^	Subscriber^		
1		AddClaim	ClaimSubscriber	Edit	Delete
2		AddClaim	OneBPSubscriber	Edit	Delete
3		TESTCSV	OneBPSubscriber	Edit	Delete

Steps and flow to implement this are as follows

1. Service

a. Read a csv file, loop through all records

b. In each loop make one json msg and add it to the json array of messages

The message structure is as follows.

Property RequestJSONArray As array Of %Text(MAXLEN = 10000);

Property Topic As %String(MAXLEN = 100);

c. End of loop

d. Send the json array to the BP

Here is the service code for reference

Class File.BS.FileServiceBS Extends Ens.BusinessService

{

Parameter ADAPTER = "EnsLib.File.InboundAdapter";

Method OnProcessInput(

```

pInput As %FileCharacterStream,
pOutput As %RegisteredObject) As %Status
{
    set DELIMITER = "^"
    set counter=1 //records read
    set JSONDynamicBPRequest = ##class(
Ent.Library.JSON.BP.JSONDynamicBPRequest).%New()
    set JSONDynamicBPRequest.Topic = "TESTCSV"
while 'pInput.AtEnd {
    set line=pInput.ReadLine()
    if ( counter > 1 ) {
        set req = ##class(%DynamicObject).%New()

        set req.ClaimId = $piece(line,DELIMITER,1)
        set req.Company = $piece(line,DELIMITER,2)
        set req.ClaimProvider = $piece(line,DELIMITER,3)
        set req.PrimaryInsuranceCarrier = $piece(line,DELIMITER,4)
        set req.EncounterDate = $piece(line,DELIMITER,5)
        $$$TRACE("req.%ToJson() "_req.%ToJson())

do JSONDynamicBPRequest.RequestJSONArray.SetAt(req.%ToJson(),
counter)
        k req
    }
    set counter=counter+1
}
do ..SendRequestSync("JSONDynamicBP",JSONDynamicBPRequest)
$$$TRACE("JSONDynamicBP loaded : " _ ( counter - 1) _
"_$ZDATETIME($ZTIMESTAMP,3,1,6))
Quit $$$OK
}

}

```

Step 2 : Business Process

- Bp gets the topic and json array of json records
- It calls the PubsubBo to get all the subscribers for this topic
- It loops through json records
- for each records. loops through the targets
- Sends the message to target via the Call activity of the BP

Notes :

- This above example can be extended and scaled as per requirement.

- Each json msg can have a topic instead of the topic being outside of array etc.
- You can use your own messages instead of json too if you need.
- You can even send messages via the pubsub bo instead of just getting targets.
(version attached)

[Download Code](#)

Any issues pls email me nv@nv-enterprises.biz

[#Microservices](#) [#Ensemble](#)

Source URL: <https://community.intersystems.com/post/pubsub-publish-subscribe-messages>