
Article

[Eduard Lebedyuk](#) · Aug 12, 2020 3m read

Calling production from a REST Broker

Productions often need to receive REST requests.

Here's how to do that.

1. Create proxy Service:

```
/// Empty BS we would use to send Produciton requests
Class production.ProxyService Extends Ens.BusinessService
{
}
```

2. Add it to production as RESTService (or any other name).

3. Write your rest broker ([docs](#), [more docs](#))

4. In your handler method create the message and call target BP/BO like this

```
Include Ensemble
Class test.REST Extends %CSP.REST
{

XData UrlMap [ XMLNamespace = "http://www.intersystems.com/urlmap" ]
{
<Routes>
<Route Url="/launchProcess/:processName" Method="POST" Call="launchProcess"/>
</Routes>
}

ClassMethod launchProcess(processName As %String) As %Status
{
    #dim sc As %Status = $$$OK

    // BS from which we would call BP
    #dim serviceName As %String = "RESTService"

    #dim message = ##class(Ens.StringContainer).%New(%request.Content.Read($$$MaxStringLength))

    set sc = ..invokeHostAsync(processName, message, serviceName)

    quit sc
}

/// Call production host in async mode
/// Thirs arg is BS caller, testing service by default - testing must be enabled in that case
ClassMethod invokeHostAsync(targetHostName As %String, message As %Persistent, servic
```

```
eName As %String = "EnsLib.Testing.Service", sessionId As %Integer = "") As %Status
{
    #dim sc As %Status

    #dim prod As Ens.Config.Production = ..getCurrentProduction()
    if '$isObject(prod) quit $$$ERROR($$$GeneralError, "Produciton is not running")

    #dim item As Ens.Config.Item = ..findConfigItem(targetHostName)
    if '$isObject(item) quit $$$ERROR($$$GeneralError, "Host " _ targetHostName _ " not found")

    #dim service As Ens.BusinessService
    set sc = ##class(Ens.Director).CreateBusinessService(serviceName, .service)
    if $$$ISERR(sc) quit sc

    if (sessionId != "") set service.%SessionId = sessionId

    set sc = service.SendRequestAsync(targetHostName, message)

    quit sc
}

/// Get current production
ClassMethod getCurrentProduction() As Ens.Config.Production
{
    #dim sc As %Status
    #dim prodName As %String
    #dim prodState As %Integer

    // Get produciton name
    if '##class(Ens.Director).IsProductionRunning(.prodName) quit ""

    // Open produciton by name
    #dim prod As Ens.Config.Production = ##class(Ens.Config.Production).%OpenId(prodName, .sc)
    if $$$ISERR(sc)
    {
        $$$LOGERROR($System.Status.GetErrorText(sc))
        quit ""
    }

    quit prod
}

/// Find host by name
ClassMethod findConfigItem(name As %String, businessType As %String = "", enabledOnly As %Boolean = 0) As Ens.Config.Item
{
    #dim sc As %Status

    // Get current production
    #dim prod As Ens.Config.Production = ..getCurrentProduction()
    if '$isObject(prod) quit ""

    // Search for our target
    #dim item As Ens.Config.Item
    #dim result As Ens.Config.Item = ""
    for i = prod.Items.Count()-1:1
```

```
{
    set item = prod.Items.GetAt(i)

    if '$isObject(item) continue
    if ((businessType '= "") && (item.BusinessType() '= businessType)) || (item.N
ame '= name) || (enabledOnly && 'item.Enabled) continue

    set result = item
    quit
}
quit result
}
}
```

And now this call:

```
POST host:port/webapp/launchProcess/BPName
{ "prop": "val" }
```

Would send a `Ens.StringContainer` message to the BPName.

Additionally REST broker can convert json to a persistent object and send that (it would be preferable).

[#Business Operation](#) [#Business Process \(BPL\)](#) [#Business Service](#) [#Interoperability](#) [#REST API](#) [#InterSystems IRIS](#)

Source URL: <https://community.intersystems.com/post/calling-production-rest-broker>