

Artículo

[Kurro Lopez](#) · 9 ago, 2021 Lectura de 4 min

[Open Exchange](#)

Añadir una configuración por defecto por código

Hola comunidad,

Este es otro artículo sobre cómo realizar acciones que puede realizar en el portal web pero mediante código.

Hoy Agregar un valor de configuración por defecto por código

Introducción

Si quieres añadir un nuevo valor por defecto, normalmente lo haces a través de l configuración de producción.

Server LAPTOP-KURRO Namespace TEST [Switch](#) User [Kurro Lopez](#) Licensed To

Welcome, Kurro Lopez

View:

Home	Configure >	Production ⓘ
	Build >	Business Partners ⓘ
	View >	Credentials ⓘ
	List >	Schedule Specs ⓘ
	Monitor >	Data Lookup Tables ⓘ
	Manage >	System Default Settings ⓘ
	Interoperate >	Purge Data Settings ⓘ
	Test >	Enterprise Systems ⓘ
Analytics		Public-Service Registry
		External-Service Registry ⓘ
		Message Bank Link ⓘ
Interoperability		
System Operation		
System Explorer		
System Administration		

System Default Settings

New

Edit

Delete

System Default Settings currently defined in namespace TEST:

Production Name	Item Name	Host Class Name	Setting Name	Setting Value	Deployable
Test.MyProduction	Test.BS.MyServiceClass	*	lang	es	No

Entonces tu configuración por defecto está asignado a tu objeto de negocio, in este ejemplo, a Test.BS.MyServiceClass

Test.BS.MyServiceClass

Settings Queue Log Messages Jobs Actions

Apply Search: []

Class Name
Test.BS.MyServiceClass

Description

Adapter Class Name

Adapter Description

Business Partner

Basic Settings

Enabled

lang
es

Additional Settings

Si no puedes acceder a la configuración por defecto en el porta, este es tu método para añadir elementos.

Nota: Para este ejemplo, he creado una clase simple %CSP.REST para usarlo como servicio API y llama a la clase [NumberTranslate](#) class. Puedes descargarlo en Open Exchange.

AddDefaultSetting

```

/// Insert or update a default value
/// <ul>
/// <li><var>pItemName</var> Set the name of item. Optional.</li>
/// <li><var>pHostClass</var> Set the name of the Host Class. Optional.</li>
/// <li><var>pSettingName</var> Set the setting name. Mandatory.</li>
/// <li><var>pSettingValue</var> Set the default value. Optional.</li>
/// <li><var>pProduction</var> Set the name of the productio, if is empty, this value
s will apply to all productions (*). Optional.</li>
/// </ul>
/// <example>
/// /// Add new default setting for the current production
/// Do myClass.AddDefaultSetting("Host.RS.Rest",,"Port","21","Test.MyProduction")
/// // Will be: Test.MyProduction|Host.RS.Rest|*|Port --> 21
/// /// Add new default setting for all productions.
/// Do myClass.AddDefaultSetting("Host.RS.Rest",,"Port","21")
/// // Will be: *|Host.RS.Rest|*|Port --> 21
/// </example>
ClassMethod AddDefaultSetting(pItemName As %String = "*", pHostClass As %String = "*"
, pSettingName As %String, pSettingValue As %String, pProduction As %String = "*") As
%Status
{
    Set ret = $$$OK

    Try {

        // Validate mandatories parameters
        If (pSettingName '=')
        {
            Set template = "%1||%2||%3||%4"
            Set configId = ##class(%Library.MessageDictionary).FormatText(template,pP
roduction, pItemName, pHostClass, pSettingName)
            If '##class(Ens.Config.DefaultSettings).%ExistsId(configId)
            {
                Set conf = ##class(Ens.Config.DefaultSettings).%New()
                Write !,"Create new config "_configId
            }
            Else
            {
                Set conf = ##class(Ens.Config.DefaultSettings).%OpenId(configId)
                Write !,"Update config "_configId
            }

            Set conf.ProductionName = production
            Set conf.ItemName = pItemName
            Set conf.HostClassName = pHostClass
            Set conf.SettingName = pSettingName
            Set conf.SettingValue = pSettingValue
            Set conf.Deployable = 1

            Do conf.%Save()
            Kill conf
        }
    }
    Else

```

```

    {
        If (pSettingName = "") set attrib = "pSettingName"

        $$$ThrowOnError($System.Status.Error(406,attrib))
    }
}
Catch ex {
    Set ret = ex.AsStatus()
    Write !,"Error #",$System.Status.GetErrorCodes(ret),!
    Write !,$System.Status.GetOneStatusText(ret,1),!
}

Quit ret
}

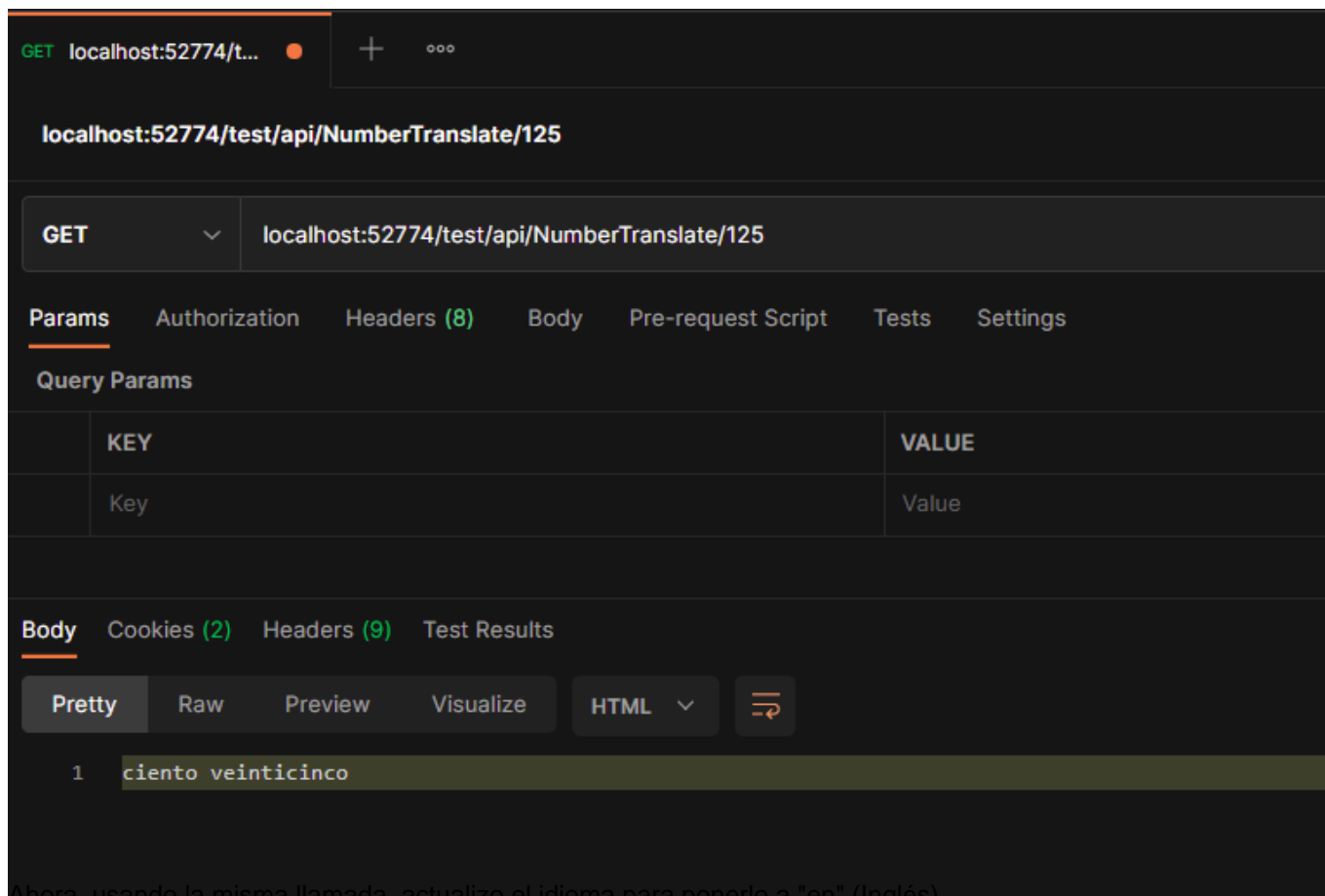
```

En este ejemplo, he usado la siguiente llamada:

```

Do myClass.AddDefaultSetting("Test.BS.MyServiceClass",,"lang","es","Test.MyProduction")

```



Ahora, usando la misma llamada, actualizo el idioma para ponerlo a "en" (Inglés)


```

Do myClass.AddDefaultSetting("Test.BS.MyServiceClass",,"lang","en","Test.MyProduction")

```

Test.BS.MyServiceClass

Settings Queue Log Messages Jobs Actions

Apply ▼  Search:

▼ **Informational Settings**

Comment


Category

Class Name

Description

Adapter Class Name

Adapter Description

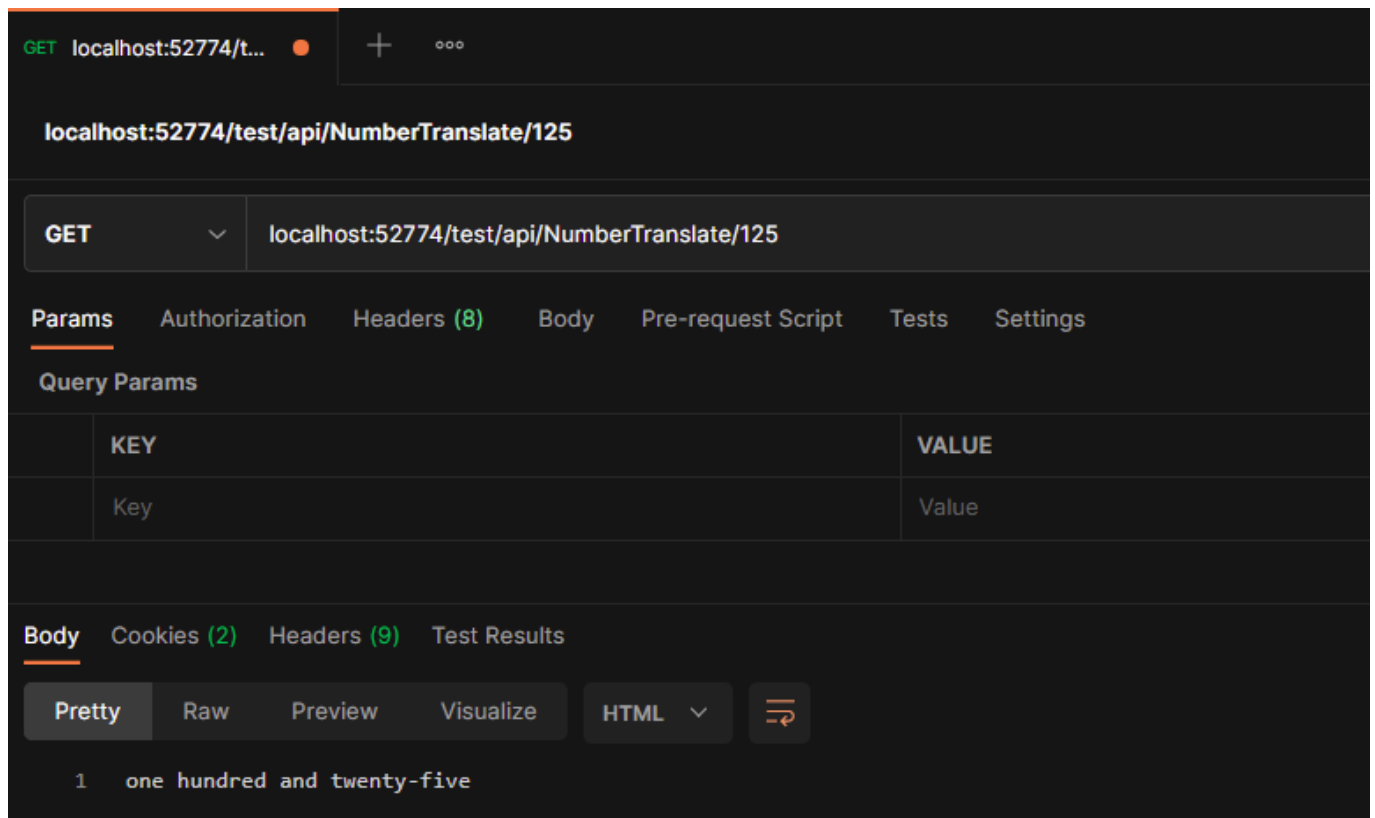
Business Partner
 

▼ **Basic Settings**

Enabled

lang

▼ **Additional Settings**



Si quiero eliminar el parámetro por defecto, este es el método.

RemoveDefaultSetting

```

/// Remove a default setting
/// <ul>
/// <li><var>pItemName</var> Set the name of item. Optional.</li>
/// <li><var>pHostClass</var> Set the name of the Host Class. Optional.</li>
/// <li><var>pSettingName</var> Set the setting name. Mandatory.</li>
/// <li><var>pProduction</var> Set the name of the productio, if is empty, this value
s will apply to all productions (*). Optional.</li>
/// </ul>
/// <example>Do myClass.RemoveDefaultSetting("Host.RS.Rest",,"Port")</example>
ClassMethod RemoveDefaultSetting(pItemName As %String = "*", pHostClass As %String =
"*", pSettingName As %String, pProduction As %String = "*") As %Status
{
    Set ret = $$$OK

    Try {

        // Validate mandatories parameters
        If (pSettingName '=')
        {
            Set template = "%1||%2||%3||%4"

            Set configId = ##class(%Library.MessageDictionary).FormatText(template,pP
roduction, pItemName, pHostClass, pSettingName)

            If ##class(Ens.Config.DefaultSettings).%ExistsId(configId)
            {
                Do ##class(Ens.Config.DefaultSettings).%DeleteId(configId)
            }
        }
    }
}

```

```
        Else
        {
            Write !,"Configure parameter not found ["_configId_]"
        }
    }
    Else
    {
        If (pSettingName = "") set attrib = "pSettingName"

        $$$ThrowOnError($System.Status.Error(406,attrib))
    }
}
Catch ex {
    Set ret = ex.AsStatus()
    Write !,"Error #",$System.Status.GetErrorCodes(ret),!
    Write !,$System.Status.GetOneStatusText(ret,1),!
}

Quit ret
}
```

Espero que estos métodos te sean útiles.

Saludos y happy coding

Kurro Lopez

[#Consejos y trucos](#) [#Portal de Administración](#) [#Principiante](#) [#Caché](#) [#Ensemble](#) [#InterSystems IRIS](#)
[Ir a la aplicación en InterSystems Open Exchange](#)

URL de

fuelle:<https://es.community.intersystems.com/post/a%C3%B1adir-una-configuraci%C3%B3n-por-defecto-por-c%C3%B3digo>