

Article
[Kurro Lopez](#) · Aug 9, 2021 4m read
[Open Exchange](#)

Add a default setting value by code

Hi community,

This is another article about how to perform actions that you can do in the web portal but via code.

Today.... Add a default setting value by code

Introduction

If you want to add a new default setting value, you usually do it from the production configuration.

Server LAPTOP-KURRO Namespace TEST Switch User Kurro Lopez Licensed To [REDACTED]

Welcome, Kurro LopezView: [Icon] [Icon]

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

Interoperability > System Default Settings

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

Then your default setting is setted in your business object, in this sample, into Test.BS.MyServiceClass

Test.BS.MyServiceClass

SettingsQueueLogMessagesJobsActions

Apply▼ Search:

Class Name

Description

Adapter Class Name

Adapter Description

Business Partner
 

▼ Basic Settings

Enabled
☒

lang

▼ Additional Settings

If you are not able to access to default setting in the portal, this is your method to add items.

Note: For this sample, I've created a simple %CSP.REST class to use as API services and it calls the

[NumberTranslate](#) class. You can download this class in 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
        {

```

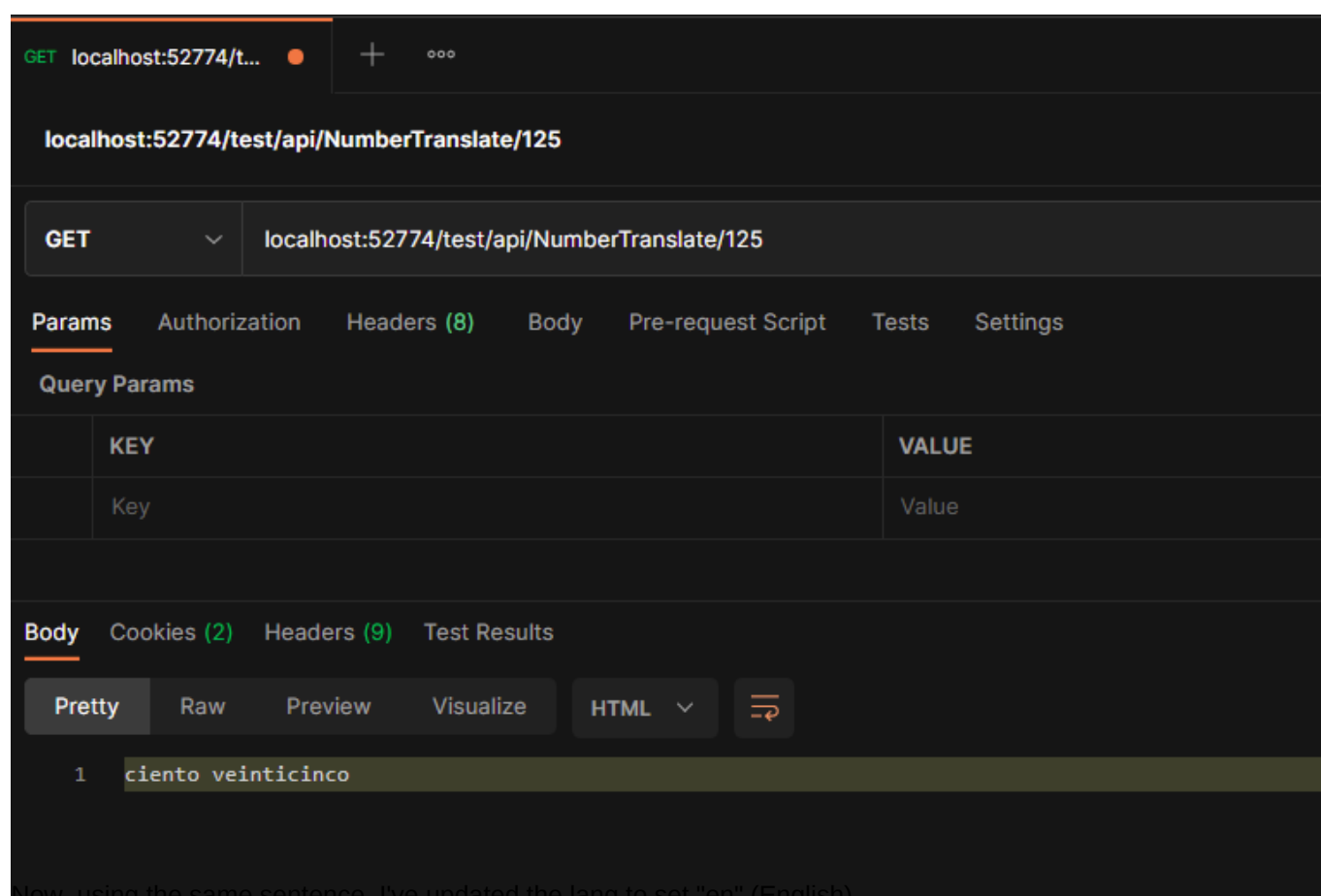
```
        If (pSettingName = "") set attrib = "pSettingName"

        $$$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
}
```

In this sample, I've used this sentence:

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



Now, using the same sentence, I've updated the lang to set "en" (English)

```
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

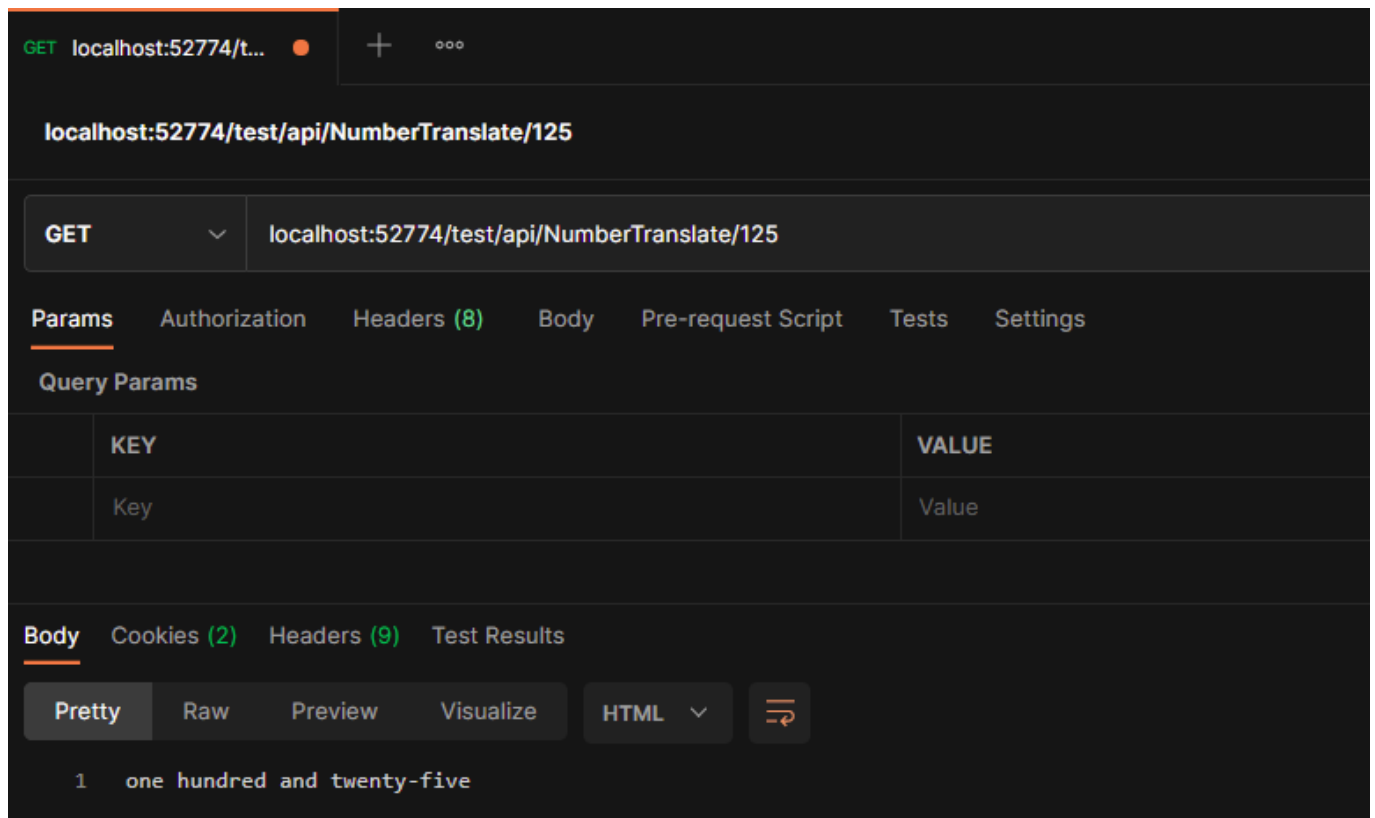
Business Partner
 

▼ Basic Settings

Enabled
☒

lang

▼ Additional Settings



If you want to remove a default setting, this is the method.

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 production, 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 mandatory 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
}
```

I hope these methods are helpful for you.

Best regards and happy coding

Kurro Lopez

[#Beginner](#) [#Management Portal](#) [#Tips & Tricks](#) [#Caché](#) [#Ensemble](#) [#InterSystems IRIS](#)
[Check the related application on InterSystems Open Exchange](#)

Source URL: <https://community.intersystems.com/post/add-default-setting-value-code>