Article
Kurro Lopez · Jul 30, 2021 3m read

Add a Web application by code

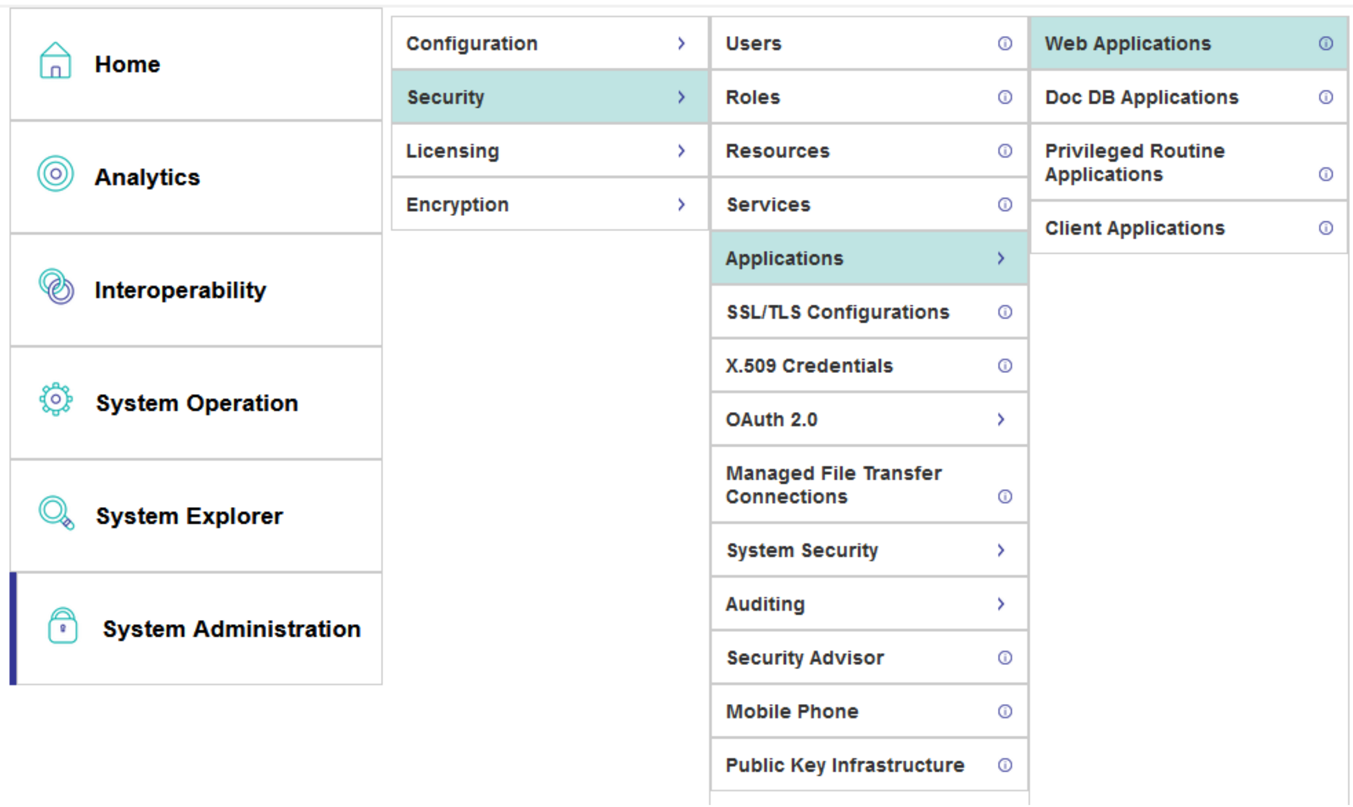
Hi community,

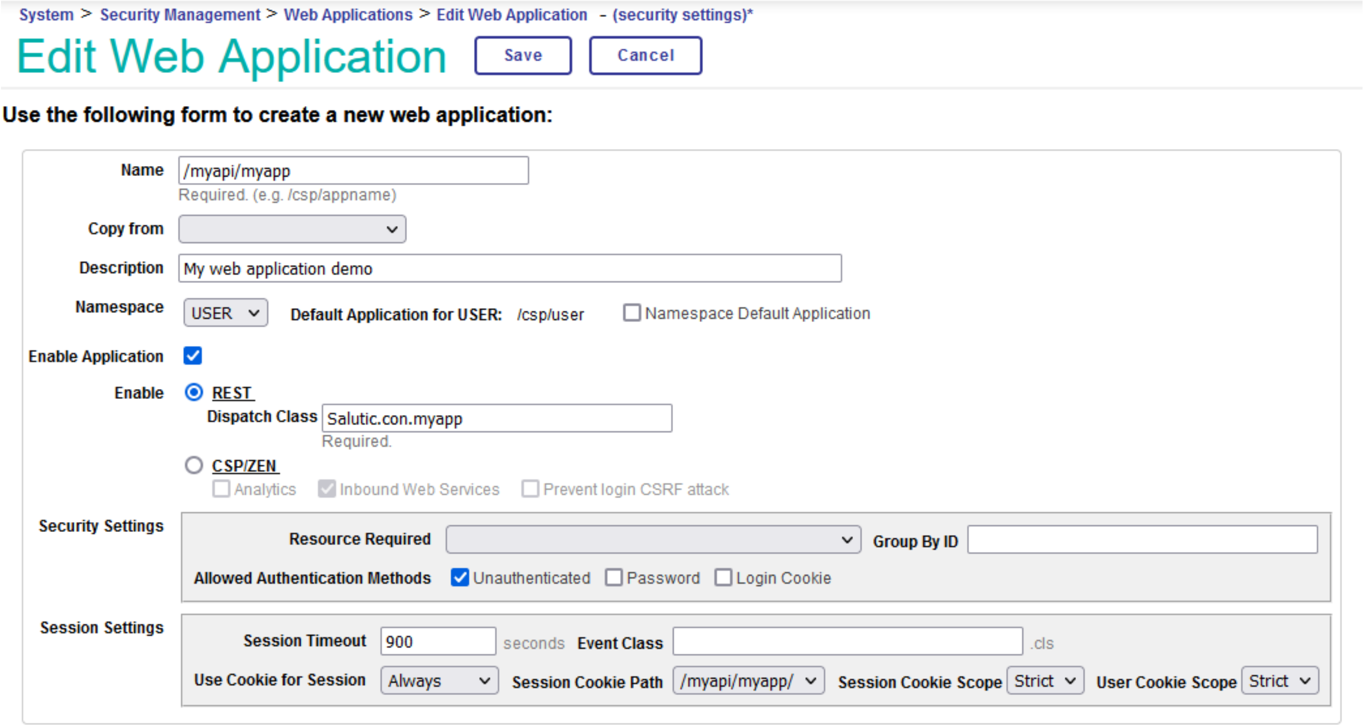
I'm going to publish several articles on how to perform actions that you can do in the web portal but via code.

Today.... Web Applications via code

Introduction

If you want to add a new Web application, you you usually do it with the option System Administration - Security - Applications - Web Applications

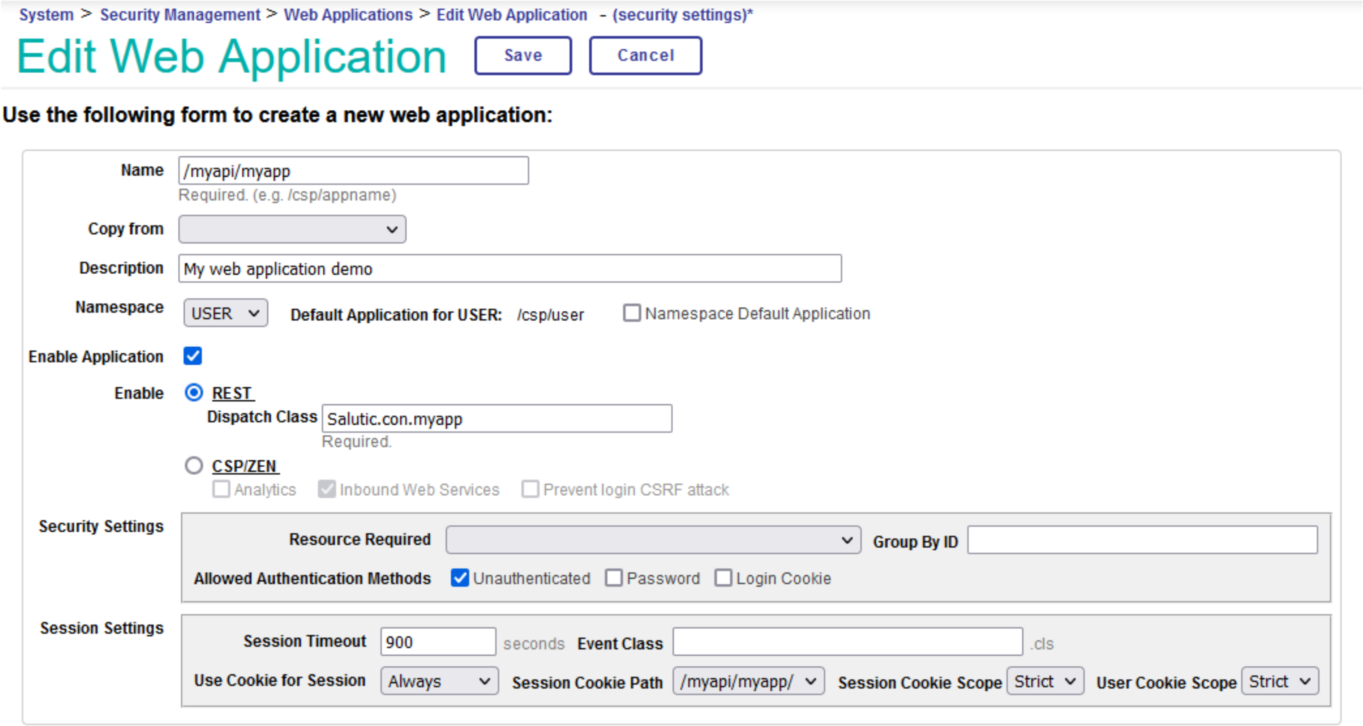




Use the following form to create a new web application:

Name: /myapi/myapp
Required. (e.g. /csp/appname)
Copy from:
Description: My web application demo
Namespace: USER
Default Application for USER: /csp/user
☐ Namespace Default Application
Enable Application: ☑
Enable: ● REST
Dispatch Class: Salutic.con.myapp
Required.
○ CSP/ZEN
☐ Analytics ☑ Inbound Web Services ☐ Prevent login CSRF attack
Security Settings
Resource Required:
Group By ID:
Allowed Authentication Methods: ☑ Unauthenticated ☐ Password ☐ Login Cookie
Session Settings
Session Timeout: 900 seconds
Event Class: .cls
Use Cookie for Session: Always
Session Cookie Path: /myapi/myapp/
Session Cookie Scope: Strict
User Cookie Scope: Strict

Then add the roles to use in the application


System > Security Management > Web Applications > Edit Web Application - (security settings)*
Edit Web Application
Save
Cancel

Use the following form to create a new web application:

Name: /myapi/myapp
Required. (e.g. /csp/appname)
Copy from:
Description: My web application demo
Namespace: USER
Default Application for USER: /csp/user
☐ Namespace Default Application
Enable Application: ☑
Enable: ● REST
Dispatch Class: Salutic.con.myapp
Required.
○ CSP/ZEN
☐ Analytics ☑ Inbound Web Services ☐ Prevent login CSRF attack
Security Settings
Resource Required:
Group By ID:
Allowed Authentication Methods: ☑ Unauthenticated ☐ Password ☐ Login Cookie
Session Settings
Session Timeout: 900 seconds
Event Class: .cls
Use Cookie for Session: Always
Session Cookie Path: /myapi/myapp/
Session Cookie Scope: Strict
User Cookie Scope: Strict

But, if you have not access to the management portal, you can do it by code.

RegisterWebApplication

With this method, a new web application can be registered via code.

```
/// Register a web application. It configure only services that doesn't need special access.
/// <ul>
/// <li><var>pNameSpace</var> Name of the namespace when the web application runs.</li>
/// <li><var>pName</var> Service name to create/update.</li>
/// <li><var>pClassName</var> Name of class to run.</li>
/// <li><var>pDescription</var> Comment to include.</li>
/// </ul>
/// <example>
/// // Add web api
/// Do myClass.RegisterWebApplication("SAMPLES","/myApp/api","Host.RS.Rest","This is my api rest")</example> ClassMethod RegisterWebApplication(pNameSpace As %String, pName As %String, pClassName As %String, pDescription As %String = "") As %Status
{
    New $Namespace
    Set $Namespace = "%SYS"
    Set ret = $$$OK
    Set spec("AutheEnabled") = $$$AutheUnauthenticated
    Set spec("NameSpace") = pNameSpace
    Set spec("Description") = pDescription
    Set spec("IsNameSpaceDefault") = $$$NO
    Set spec("DispatchClass") = pClassName
    Set spec("MatchRoles")=":%All"

    If ('##class(Security.Applications).Exists(pName)) {
        Write !,"Creating Web application """_pName_"""..."
        Set ret = ##class(Security.Applications).Create(pName, .spec)
        Write !, "Web application """_pName_""" is created."
    }
    Else { // ensure configuration matches in case of updating from old terminal versions
        Write !, "Updating web application """_pName_"""..."
        Set ret = ##class(Security.Applications).Modify(pName, .spec)
        Write !, "Web application """_pName_""" is updated."
    }
    Return ret
}
```

By default, it uses the %All roles, but you can modify according your needs

RemoveWebApplication

```
/// Remove a web application
/// <ul>
/// <li><var>pName</var> Name of web application to remove.</li>
/// <li><var>pClassName</var> Name of the class linked to the web application.</li>
/// </ul>
/// <example>
/// // Remove web api
/// Do myClass.RemoveWebApplication("/myApp/api","Host.RS.Rest")</example>
ClassMethod RemoveWebApplication(pName As %String, pClassName As %String)
{
    New $Namespace
    Set $Namespace = "%SYS"
    Set ret = $$$OK
    If (##class(Security.Applications).Exists(pName)) {
        Do ##class(Security.Applications).Get(pName, .props)
        If (props("DispatchClass") '= pClassName) {
            Write !, "Web application doesn't refer to DispatchClass "_pClassName
        }
        Else {
            Write !, "Deleting Web application """_pName_"""..."
            Set ret = ##class(Security.Applications).Delete(pName)
            Write !, "Web application """_pName_""" was successfully deleted."
        }
    }
    Return ret
}
```

For security, the name of the class is necesary to check that you are not removing a web application by error.

I hope it helps you.

Best regards,
Kurro Lopez

#API #Beginner #Code Snippet #ObjectScript #Tips & Tricks #Caché #Ensemble #InterSystems IRIS

Source URL:https://community.intersystems.com/post/add-web-application-code