Article

Muhammad Waseem · Nov 29, 2021 3m read

Open Exchange

# OAuth2 and Basic Authentication, Authorization AND Auditing by code from Web Application

In this article I will explain how to Authenticate, Authorize and Audit by code by using CSP Web Application along with Enabling /Disabling and Authenticate/Unauthenticate any Web Application.

- Online Demo is available on cloud <a href="https://dappsecurity.demo.community.intersystems.com/csp/user/index.csp">https://dappsecurity.demo.community.intersystems.com/csp/user/index.csp</a> (SuperUser | SYS)
- I recommend to watch the video before continue <a href="https://www.youtube.com/watch?v=qFRa3njqDcA">https://www.youtube.com/watch?v=qFRa3njqDcA</a>

#### **Application Layout**

Management Portal Web Terminal Main Tasks Data Form Login	
---	--

User: \_SYSTEM

Roles: %AII,%DB\_USER

Create TestUser
Grant Read/Write Access
Grant All Privileges

Disable WebTerminal Application

Enable WebTerminal Application

Disable WebTerminal Authentication

Enable WebTerminal Authentication

## Let's start with Authentication

Authentication verifies the identity of any user or other entity attempting to connect to InterSystems IRIS®. As it 's often said, authentication is how you prove that you are who you say you are.

There are a number of different ways that a user can be authenticated; each is known as an authentication mechanism. InterSystems IRIS supports a number of authentication mechanisms:

- <u>Kerberos</u> The Kerberos protocol was designed to provide secure authentication to services over an
  unsecured network. Kerberos uses tickets to authenticate a user and avoids the exchange of passwords
  across the network.
- Operating System-Based OS-based authentication uses the operating system 's identity for each user to identify that user to InterSystems IRIS.
- <u>Instance Authentication</u> With Instance authentication, InterSystems IRIS prompts the user for a password and compares a hash of the provided password against a value it has stored.
- <u>Lightweight Directory Access Protocol (LDAP)</u> With the Lightweight Directory Access Protocol, InterSystems IRIS authenticates the user based on information in a central repository, known as the LDAP server.
- <u>Delegated Authentication</u> Delegated authentication provides a means for creating customized authentication mechanisms. The application developer entirely controls the content of delegated authentication code.

I am using Instance Authentication, for User creation we can use following objectscript command:

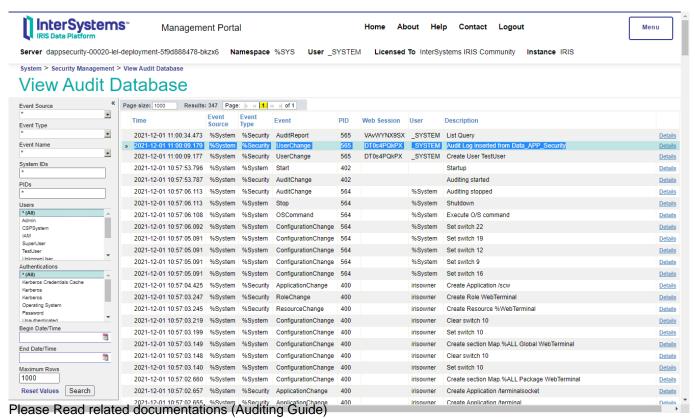
```
&sql(CREATE USER TestUser IDENTIFY BY demo)
```

We created TestUser with demo password

## **Auditing**

Upon creating user record is also adding in auditing database by using below objectscript command:

Do \$SYSTEM.Security.Audit("%System","%Security","UserChange","User:TestUser | Passwor d:demo","Audit Log inserted from Data\_APP\_Security")



: https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls?KEY=AAUDIT

### **Authorization**

Once authentication is done we need to create roles and grant Privileges to the roles and then link roles with users (<u>Authorization</u>). This we will do in three steps

Step 1: Create Role by using following objectscript command, We are creating ReadWrite role

```
&sql(CREATE ROLE ReadWrite)
```

Step 2 : Grant SELECT, UPDATE, INSERT Privileges ON table to the Role, We are assigning scw. Patient table privileges to ReadWrite role

```
&sql(GRANT SELECT, UPDATE, INSERT ON scw. Patient TO ReadWrite)
```

Step 3: Grant Role to the User, We are assigning ReadWrite role to TestUser user

```
&sql(GRANT ReadWrite To TestUser)
```

# Enable/Disable web application

We can enable or disable web application by using following objectscript code

```
New $Namespace
Set $Namespace = "%SYS"
Set App = ##class(Security.Applications).%OpenId("/terminal")
```

OAuth2 and Basic Authentication, Authorization AND Auditing by code from Web Application Published on InterSystems Developer Community (https://community.intersystems.com)

```
Set App.Enabled=0
Do App.%Save()
```

here "/terminal" is the name of our application. Application can be disable by setting "App.Enabled" to 0 and enable by setting value to 1

## Authenticate/Unauthenticate Web application

We can set Authentication by using following objectscript code

```
New $Namespace
Set $Namespace = "%SYS"
Set App = ##class(Security.Applications).%OpenId("/terminal")
Set App.AutheEnabled=0
Do App.%Save()
```

here "/terminal" is the name of our application. Authentication can be set by using "App.AutheEnabled" property. Following numeric values can be set

```
property AutheEnabled as Security.Datatype.Authentication [ InitialExpression = 64 ];
Authentication and Session mechanisms enabled (CSP Only).
Bit 2 = AutheK5API
Bit 5 - AutheCache
Bit 6 = AutheUnauthenticated
Bit 11 = AutheLDAP
Bit 13 = AutheDelegated
Bit 14 = LoginToken
Bit 20 = TwoFactorSMS
Bit 21 = TwoFactorPW
```

#### Thanks

#Authentication #Best Practices #OAuth2 #InterSystems IRIS Check the related application on InterSystems Open Exchange

#### Source

 $\label{lem:url:https://community.intersystems.com/post/oauth2-and-basic-authentication-authorization-and-auditing-code-web-application$