

Artigo

[Yuri Marx](#) · Nov. 30, 2021 3min de leitura

[Open Exchange](#)

## O poder do XDATA aplicado à segurança da API

O XData (<https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls?KEY=GOBJXDATA>) é um recurso poderoso para definir informações de documentação e metadados para classes e métodos. A classe %CSP.REST usa XDATA para mapear chamadas REST (<https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls?KEY=GRESTcsprest>), portanto, neste artigo, você verá como usar XData em seus aplicativos como código, não apenas como documentação.

Quando você escreve comentários / definições XData, o IRIS os armazena em %Dictionary.ClassDefinition (para classes) e %Dictionary.MethodDefinition (para métodos). Se você consultar essas tabelas, poderá obter informações de metadados e escrever código para essa configuração de metadados. %CSP.REST faz isso quando você escreve seus mapeamentos REST para seus serviços REST usando ObjectScript.

Eu escrevi um aplicativo que está usando XDATA para impor regras de autorização aos endpoints do método de classe. consulte:

```
/// Retrieve all the records of dc.Sample.Person
/// @security.and: roles: { PersonAdmin }
ClassMethod GetAllPersons() As %Status
{
    #dim tSC As %Status = $$$OK
    ....
}
```

O @ security.and não existe no IRIS. Portanto, preciso ler essa configuração e escrever o código para garantir o acesso aos usuários apenas com a role PersonAdmin.

Para obter este @ security.and, você precisa ler este XData. Veja:

```
ClassMethod GetXDataContent(className, methodName) As %String
{
    Set qryXdata = "SELECT parent, Name, Description FROM
%Dictionaty.MethodDefinition WHERE parent = ? and Name = ?"
    Set stmXdata = ##class(%SQL.Statement).%New()
    Set qStatus = stmXdata.%Prepare(qryXdata)
    If qStatus'=1 {Write "%Prepare
failed:" Do $System.Status.DisplayError(qStatus) Quit}
    Set rsetXdata = stmXdata.%Execute(className, methodName)
    While rsetXdata.%Next() {
        // Return rsetXdata.Name
    }
    Return rsetXdata.Description
}
```

```

}
}

```

Com esse método, você pode obter qualquer conteúdo xdata para métodos.

Agora, para restringir o acesso apenas aos usuários com a role PersonAdmin é simples, você precisa fazer o override do método de classe AccessCheck da classe %CSP.REST. Veja:

```

ClassMethod AccessCheck(Output pAuthorized As %Boolean = 0) As %Status
{
    Do ##super()

    Set message = {}

    Set tSC = $$$OK

    Set message.verb = %request.Method
    Set message.url = %request.URL
    Set message.url = "/"_$(REPLACE(message.url, %request.Application, ""))
    Set message.application = %request.Application

    Set methodName = ""
    Do ..GetClassName(message.url, %request.Method, .methodName)

    Set message.method = methodName
    Set xdata = ##class(dc.SecurityMediator.XDataUtil).
    GetXDataContent($CLASSNAME(), methodName)
    Do ..GetSecurityRules(xdata, .rules, .roles, .header, .operator)
    Set UserRoles = $LISTFROMSTRING($ROLES, ",")
    Set RolesAllowed = UserRoles

    If $FIND(xdata, "@security") > 0 {
        Set RolesAllowed = $LISTFROMSTRING(roles, ",")
    }

    Set HasRole = 0

    For RoleIdx=1:1:$LISTLENGTH(UserRoles) {
        If $LISTFIND(RolesAllowed, $LIST(UserRoles, RoleIdx)) {
            Set HasRole = 1
        }
    }
}

```

```
}  
  If HasRole {  
    Set pAuthorized = 1  
  } Else {  
    Set pAuthorized = 0  
    Set message.error = $USERNAME_" is not authorized for this request. User  
Roles Allowed is not in User Roles"  
    Write message.%ToJSON()  
  }  
  Return tSC  
}
```

Se a regra corresponder, é setado pAuthorized = 1, caso contrário, é setado 0.

Agora, as funções permitidas são baseadas na configuração de XData para sua classe REST. Excelente!

Para meu novo app em ação vá em: <https://openexchange.intersystems.com/package/API-Security-Mediator>.

[#ObjectScript](#) [#REST API](#) [#Segurança](#) [#InterSystems IRIS](#)  
[Confira o aplicativo relacionado no InterSystems Open Exchange](#)

---

URL de  
origem: <https://pt.community.intersystems.com/post/o-poder-do-xdata-aplicado-%C3%A0-seguran%C3%A7a-da-api>