
Question

[Yone Moreno](#) · Jan 25, 2021

How could we retrieve an image from an external system using Ensemble?

Hello,

We would be grateful if you could help us,

Our aim is to develop a method which retrieves an image stored in an external system, using a REST Operation

To upload the image we have the following code:

```
/// Operacion para gestionar el Registro imagen identidad digital en Tarjeta Sanitaria
Class Operaciones.REST.miSCS.miSCS Extends EnsLib.REST.Operation
{

Parameter INVOCATION = "Queue";

/// Subir la imagen del usuario
/// La documentación para crear este metodo es:
/// https://docs.intersystems.com/irisforhealthlatest/csp/docbook/Doc.View.cls?KEY=GET_http#GNET_http_post_multipart
/// https://docs.intersystems.com/irisforhealthlatest/csp/docbook/Doc.View.cls?KEY=GET_mime
Method SubirImagen(pRequest As Mensajes.Request.miSCS.SubirImagen, pResponse As Mensajes.Response.miSCS.SubirImagen) As %Library.Status
{
$ $$LOGINFO("pRequest.imagen: " _pRequest.imagen) //Creamos Request y Response HTTP
Set httpRequest=##class(%Net.HttpRequest).%New()
set tResponse = ##class(%Net.HttpResponse).%New() // Generamos la parte que contiene
a las demás partes, la "raiz", RootMIMEPart
Set RootMIMEPart=##class(%Net.MIMEPart).%New()

//Creamos la primera "subparte" con el expediente
Set ExpedienteMIMEPart=##class(%Net.MIMEPart).%New()
Set contentdisp="form-data; name=""expediente"""
Do ExpedienteMIMEPart.SetHeader("Content-Disposition",contentdisp)
Set ExpedienteMIMEPart.Body=pRequest.expediente
//Do ExpedienteMIMEPart.SetHeader("Content-
Type", "text/plain") ; esto no se envia en POSTMAN

//Asignamos la imagen
//Esto si funciona (la guarda como jpg y al descargarla se ve)
Set docStream=##class(%Stream.FileBinary).%New()
//Set sc=docStream.LinkToFile("C:\InterSystems\avatar prueba.jpg") // Esto no
funciona (guarda el base64 directo, no el binario;
// es decir, guarda lo mismo que enviamos desde POSTMAN)
```

```
//set docStream = pRequest.imagen

//Si ponemos pRequest.imagen como string, enviamos el base64 por postman y decodificamos aqui a binario si
//guarda la imagen como jpg
set binary = $system.Encryption.Base64Decode(pRequest.imagen)
do docStream.Write(binary)

// Instanciamos la segunda "subparte" con la imagen
Set ImagenMIMEPart=##class(%Net.MIMEPart).%New()
// Incluimos la imagen, el docStream, en la subparte MIME
Do ImagenMIMEPart.BodySet(docStream)
// Cabeceras
Set ImagenMIMEPart.ContentType = "image/jpeg"
Do ImagenMIMEPart.SetHeader("Content-Disposition", "form-data; name=""imagen""; filename=""imagen.jpg""")

// Insertamos las dos subpartes en la raiz
Do RootMIMEPart.Parts.Insert(ExpedienteMIMEPart)
Do RootMIMEPart.Parts.Insert(ImagenMIMEPart) // Creamos el MIME writer
Set writer=##class(%Net.MIMEWriter).%New() // Preparamos para enviar el mensaje MIME a la peticion HTTP
Set httpRequest=##class(%Net.HttpRequest).%New()
Set status=writer.OutputStream(httpRequest.EntityBody)
if $$$ISERR(status) {do $SYSTEM.Status.DisplayError(status) Quit} // Escribimos el contenido del mensaje MIME en la peticion HTTP
Set status=writer.WriteMIMEBody(RootMIMEPart)
if $$$ISERR(status) {do $SYSTEM.Status.DisplayError(status) Quit} // Creamos la URL
set url      = ##class(Util.TablasMaestras).getValorMaestra("MISCS.PARAMETRIZACIONES","url")
set path      = ##class(Util.TablasMaestras).getValorMaestra("MISCS.PARAMETRIZACIONES","path")
set servicio = ##class(Util.TablasMaestras).getValorMaestra("MISCS.PARAMETRIZACIONES","servicio")
set recurso  = ##class(Util.TablasMaestras).getValorMaestra("MISCS.PARAMETRIZACIONES","subirImagen") set URL      = "http://" _url_path_servicio_recurso

$$$LOGINFO("URL: "_URL)
//Escribimos el tipo de contenido que enviamos en la raiz
Set httpRequest.ContentType="multipart/form-data; boundary=_RootMIMEPart.Boundary
$$$LOGINFO("> httpRequest.ContentType: "_httpRequest.ContentType) //Enviamos al sistema externo
set tSC=httpRequest.Post(URL,0)
$$$LOGINFO("tSC: "_tSC) //Lanzamos excepcion si hubo error
if $$$ISERR(tSC){
$$$ThrowOnError(tSC)
}
//Obtenemos respuesta
set tResponse = httpRequest.HttpResponse //Leemos respuesta
if (tResponse.Data.AtEnd = 0) {
set linea = tResponse.Data.Read()
$$$LOGINFO("Linea: "_linea)
} //Rellenamos response
set pResponse = ##class(Mensajes.Response.miSCS.SubirImagen).%New()
set pResponse.resultado = 1
set pResponse.informacion = linea
//Traducimos codigo del resultado, a descripcion
```

```
//set pResponse.informacion = ##class(Util.TablasMaestras).getValorMaestra("MISCS.CODIGOSTSI", linea) Quit pResponse
}

XData MessageMap
{
<MapItems>
<MapItem MessageType="Mensajes.Request.miSCS.SubirImagen">
<Method>SubirImagen</Method>
</MapItem>
</MapItems>
}
```

We have written the following code to retrieve the image:

```
/// Operacion para gestionar el la imagen
Class Operaciones.REST.miSCS Extends EnsLib.REST.Operation
{

    /// Obtener la imagen del usuario
    Method ConsultarImagen(pRequest As Mensajes.Request.miSCS.ConsultarImagen, pResponse
        As Mensajes.Response.miSCS.ConsultarImagen) As %Library.Status
    {
        set httpRequest = ##class(%Net.HttpRequest).%New()
        set tResponse = ##class(%Net.HttpResponse).%New()
        set httpRequest.ContentType = "application/json"
        set url      = ##class(Util.TablasMaestras).getValorMaestra("MISCS.PARAMETRIZACIONES"
            , "url")
        set path      = ##class(Util.TablasMaestras).getValorMaestra("MISCS.PARAMETRIZACIONES"
            , "path")
        set servicio = ##class(Util.TablasMaestras).getValorMaestra("MISCS.PARAMETRIZACIONES"
            , "servicio")
        set recurso  = ##class(Util.TablasMaestras).getValorMaestra("MISCS.PARAMETRIZACIONES"
            , "consultarImagen") set URL = "http://" _url_path_servicio_recurso

        $$$LOGININFO("URL: "_URL) set tFormVarNames = "expediente"
        set tData("expediente") = pRequest.expediente set tSC = ..Adapter.SendFormDataArray(.tResponse, "GET", httpRequest, .tFormVarNames, .tData, URL)
        $$$LOGININFO("tSC: "_tSC) if $$$ISERR(tSC){
            $$$ThrowOnError(tSC)
        }
        if (tResponse.Data.AtEnd = 0) {
            set linea = tResponse.Data.Read()
            $$$LOGININFO("Linea: "_linea)
        } set pResponse = ##class(Mensajes.Response.miSCS.ConsultarImagen).%New()
        set pResponse.resultado = 1
        set pResponse.informacion = linea Quit pResponse
    }

    XData MessageMap
```

```
{  
<MapItems>  
  <MapItem MessageType="Mensajes.Request.miSCS.ConsultarImagen">  
    <Method>ConsultarImagen</Method>  
  </MapItem>  
</MapItems>  
}
```

1 To upload an image we do the following

- We choose a JPG and open it with Notepad, we observe: ÿØÿà JFIF...
- We encode it to base64 using the following converter: <https://www.motobit.com/util/base64-decoder-encoder.asp>, the result is: /9j/4CAQSk...

-2 If we send the request to our ESB REST Service using POSTMAN to the patient 9999

The screenshot shows a POSTMAN interface with the following details:

- Method: POST
- URL: http://localhost:19621/sanidad/scs/test/miSCS/v01r00/subirImagen
- Headers tab is selected, showing:
 - request (checkbox checked)
 - Key (checkbox checked)
- Body tab is selected, showing a JSON response:

```
{"idApp": "miHistoria", "usuario": "02196228M", "expediente": "9999", "Imagen": "/9j/4AAQSkZJRgABAQAAAQABAAD/2wCEAAkGBxMSEhUTExlWFhUWGBgbGRcYFxoYGhgXGBYYFxUbGB0YHSggGholGxUVITEhjSkrLi4uFx8zODMsNygtLisBCg"}
```

The external system replies that it has been successfully uploaded

If we get this image directly from the external system endpoint, we get it as a correct JPG:

How could we retrieve an image from an external system using Ensemble?

Published on InterSystems Developer Community (<https://community.intersystems.com>)

The screenshot shows a POSTMAN interface. At the top, it says "Consultar Imagen". Below that, the method is set to "GET" and the URL is "http://tsws-pre.scs-salud.org/servicios/imagenidscs/consultarImagen". Under the "Body" tab, there is a "form-data" section with a key "expediente" and a value "9999". The "Headers" tab shows "Content-Type: application/x-www-form-urlencoded". The "Test Results" tab shows a status of "200 OK" with a response size of "9.36 KB". The response body is a large binary file (image) starting with "data:image/jpeg;base64," followed by a long string of encoded data.

It is correct because if we save it and open, it loads the original image:



3 However here comes the challenge:

When we execute the method "consultarImagen" to get the uploaded image from the ESB, to the patient 9999:

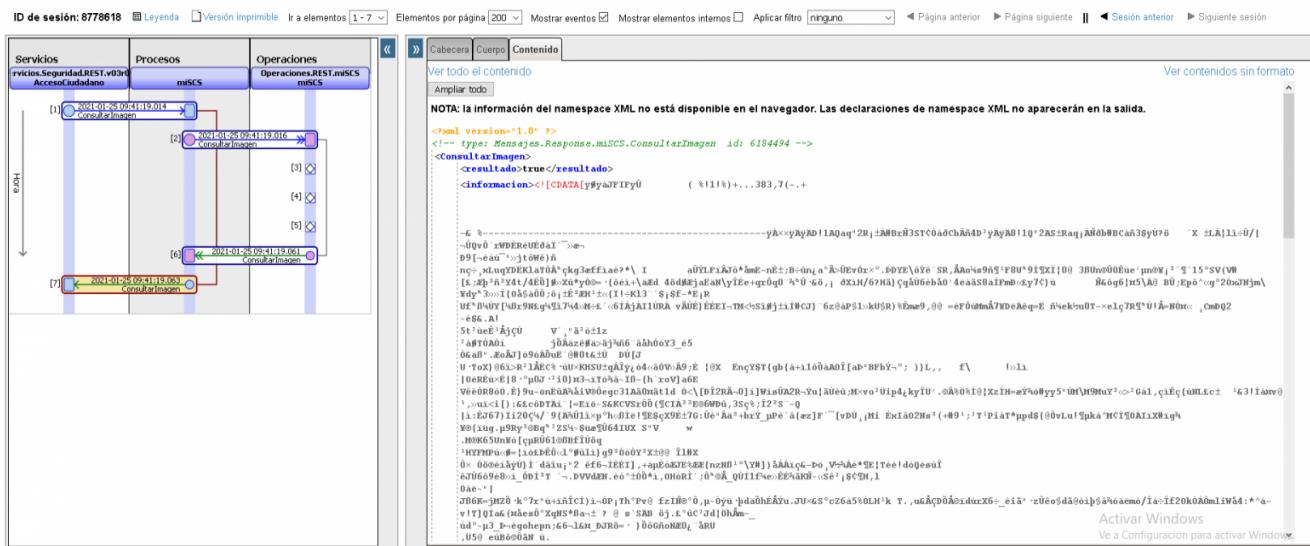
1 The external system replies something which looks like a JPG:

How could we retrieve an image from an external system using Ensemble?

Published on InterSystems Developer Community (<https://community.intersystems.com>)



And we send to the service something which could look like the original JPG:



But when we see the output being sent to the POSTMAN, we see some strange symbols like "x00" or "x01"



So if we save this response:

And we leave just the image part:

It does not represent an image



4 In addition, we have tried in our REST Service, to convert the image to base64 as follows:

```

/// Obtener la imagen guardada
Method consultarImagen(pInput As %Stream.Object, Output pOutput As %Stream.Object) As
%Status
{
Set pOutput=##class(%GlobalBinaryStream).%New()
set claseAux = ##class(%ZEN.Auxiliary.jsonProvider).%New() //Necesitamos sacar la cab
ecera "request" de la peticion REST
set req = $tr(pInput.GetAttribute("request"), "")
set tSC= claseAux.%ConvertJSONToObject(.req, "Mensajes.Request.miSCS.ConsultarImagen",
.objetoEntrada,1) //Enviamos al Proceso
set tSC = ..SendRequestSync("miSCS",objetoEntrada,.objetoSalida)

//Implementar el codificar la imagen que viene en objetoSalida.informacion de texto c
on letras extrañas a base64 $$$LOGINFO("objetoSalida.informacion: "_objetoSalida.info
rmacion)
set imagenBase64 = $System.Encryption.Base64Encode(objetoSalida.informacion)
$$$LOGINFO("imagenBase64: "_imagenBase64)
set objetoSalida.informacion = imagenBase64
$$$LOGINFO("objetoSalida.informacion: " objetoSalida.informacion)

```

```
//Convertimos el OBJETO devuelto por el Proceso en JSON
set tSC = claseAux.%WriteJSONStreamFromObject(.pOutput,.objetoSalida,,,,"aeloqtuw") /
/Enviamos el JSON con cabeceras
Do:$$$$ISOK(tSC) pOutput.SetAttribute("Content-Type", "application/json")
do pOutput.SetAttribute("Access-Control-Allow-Origin", "*")
do pOutput.SetAttribute("Access-Control-Allow-Credentials", "true")
do pOutput.SetAttribute("Access-Control-Allow-Methods", "GET")
    do pOutput.SetAttribute("Access-Control-Allow-Headers", "request,Access-Control-Allow-Header, Origin,Accept, X-Requested-With, Content-Type, Access-Control-Request-Method, Access-Control-Request-Headers")
Quit tSC
}
```

If we get the image now, we observe it as base64:

The screenshot shows a browser's developer tools Network tab with a single entry for 'image'. The response status is 200 OK, time is 896 ms, size is 13.6 KB, and the content type is image. The response body is a large base64 encoded string starting with '2wCEAAkGBxMSEhUTExiWFhUWGBgbGRcYF'. The browser interface includes tabs for Body, Cookies, Headers (8), Test Results, and a dropdown for Pretty, Raw, Preview, Visualize, and JSON.

When we download it:

```
1 {{"resultado": true,
2 "informacion": "\/9j\/4AAQSkZJRgABAQAAAQABAAAD",
3 "error": null
4 }
5 }
```

A red arrow points to the 'informacion' field in the JSON object, highlighting the base64 encoded image data.

Get just the image part:

```
1 \/9j\/4AAQSkZJRgABAQAAAQABAAAD\
```

The image string is shown in a monospaced font, indicating it's a single line of code.

Then decode it from base64:

The Form.SizeLimit is 10000000bytes. Please, do not post more data using this form.

Source data from the Base64 string:

Type (or copy-paste) some text to a textbox bellow. The text can be a Base64 string to decode or any string to encode to a Base64.

```
\v9j\4AAQSkZJRgABAQAAAQABAAAD\2wCEAAkGBxMSEhUTeXIWFhUWGBgbGRcYFxoYHgXGBYYFxUb
\r\nGBoYHSggGho1GxUVITEhJSkrLi4uFx8zODMsNygtLisBCgoKDg0OGhAQGy0mICUtLS0tLS0tLS0t
\r\nLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLf
\AABEIANcA1wMBEQACEQED\r\nEQH\xAaabAAABBQEBAAAAAAAAAAAgMEBQYBB
\//EAEQQAAIBAgMEBgYGCAUFAQAAAAECAMR
\r\nBCExBRJBQYTYXGBkSIyUqGxwRQjQnKS0TNTVIKi0uHwFkNiwwEHNESTshX
\xAAaAQEEAgMBAAAA\r\nAAAAAAQAQIDBAUG|
\8QAOBEEAgECAwQHBwIHAQEAAAAAAECAXEEITESE1GRBSIyQVOBsVJh
\r\ncaHB0fBikhQVI0JD4fEzJP
\aaAwDAQACEQMRAD8A9Qm0WCAEAIAQBrFMwRymbBTu99svfKzbUXbU
\r\nntBjyV9DIUmdVcnw4c+0r7vmrA2PjNCGOVusjengHfqvIrtqdIstTiPRX6inxCm7n97h4THVxj11
E\r\ny0sFbORU0sKq52ueZzPm2pObZuxhGOg
\KlwqBAOEsoIjYdsZEZjdTEbvrzfcSo30KuVtRS1uyLE7
```

or select a file to convert to a Base64 string.

Examinar... No se ha seleccionado ningún archivo.

Convert the source data

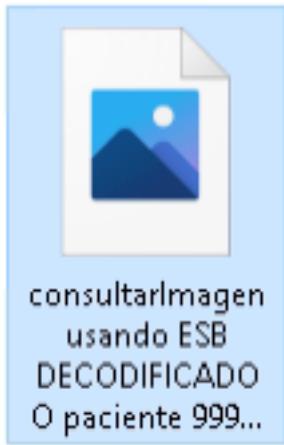
What to do with the source data?

- encode** the source data **to a Base64 string** (base64 encoding)
Maximum characters per line: 76
 - decode** the data **from a Base64 string** (base64 decoding)

Output data

- output to a textbox (as a string)
 - export to a binary file, filename: base64.bin

If we save the string generated as a jpg, it does not represent an image:



Could you help us??

We have also read:

<https://community.intersystems.com/post/decode-base64-file>

<https://community.intersystems.com/post/how-save-png-image-cache-base64-...>

<https://community.intersystems.com/post/send-image-multi-part-file>

<https://docs.intersystems.com/irisforhealthlatest/csp/docbook/Doc.View.c...>

<https://docs.intersystems.com/irisforhealthlatest/csp/docbook/Doc.View.c...>

Thanks for your help,

#Code Snippet #REST API #Ensemble

Source
URL:<https://community.intersystems.com/post/how-could-we-retrieve-image-external-system-using-ensemble>