Stream Binary to JSON work around. HealthShare 2017.2

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

Question

Ivan Novo · Feb 28, 2019

Stream Binary to JSON work around. HealthShare 2017.2

Hi Dev Community,

My team is currently working on a project where we need to post pdf documents to a 3rd party REST API.

The API specifies a field in the JSON request message to contain the 'physical content' of the file as an array of bytes, example of the JSON request as follows - data truncated for readability purposes:

```
{
  "CaptureSource":2,
  "RecipientID":"ID34",
  "Document": {
  "Guid":"5D847A4E9CC1485382DC4A5F5DC80D6C",
  "EventDate":"2019-01-14T13:23:46",
  "FileExtension":"pdf",
    "FileContent":"JVBERiOxLjQNCiXi48/TDQoxIDAgb2JqDQogIDw8DQogICA......"

  },
  "ActionRequired":false,

}
```

COB projects array of bytes as %Stream.GlobalBinary stream type. We are able to see the physical contents of the files within our production when we read the files and they travel through the business hosts in our production, but when we call the utility method:

```
Set tSC=##class(Ens.Util.JSON).ObjectToJSONStream(pRequest,.tJsonPayload,"iuw")
```

The output blanks out the FileContent element of the JSON message - pRequest being a custom request object with a Property FileContent As %Stream.GlobalBinary , and crashes the API call:

```
{
  "CaptureSource":2,
  "RecipientID":"ID34",
  "Document": {
  "Guid":"5D847A4E9CC1485382DC4A5F5DC80D6C",
  "EventDate":"2019-01-14T13:23:46",
  "FileExtension":"pdf",
  "FileContent":null
  },
```

I've tried creating a %DynamicObject instead and calling %To.JSON(), but apparently Binary Streams are not supported:

```
W !, tDynamicObj.%ToJSON()

^
<METHOD NOT SUPPORTED>zTestByteArray+9^UCLH.SIP.Test.Utilities.1 *%ToJSON,%Stream.FileBinary
```

Stream Binary to JSON work around. HealthShare 2017.2

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

We have contacted the vendor of the API to double check if they are able to process an E64 encoded version of the file content, but they are expecting it as a byte array so our choices here are zero to none. If we use POSTMAN and copy across the contents of the file into a JSON request the call works like a charm.

Is anybody aware of any workaround available? I have tried to output the %Stream.GlobalBinary as a String but it logically outputs the content of the pdf file a a stream of characters:

"FileContent":"%PDF-1.4\r\n%Ã¢Ã£ÃÃ\r\n1 0 obj\r\n <<\r\n \/Author()\/Title()\/Subject()\/Producer()\/Keywords()\/Cr eationDate(D:20190114132345+00'00')\/ModDate(D:20190114132345+00'00')\/Creator(CSSystems Corporation)\r\n >>\r\nendobj\r\n4 0 obj\r\n <<\/Filter \/FlateDecode \/Length 1314>>\r\n stream\r\nxÂÃ- ÂmoÃ6ÂÂ¿ÃÂọmÂ4!%R\"Âa@ ....

#Development Environment #JSON #ObjectScript #REST API #HealthShare