

# How do I return binary media in my API?

In our [What is a resource?](#) article we noted that a resource is digital content which can be represented by various media types. Most of the time this will be text based media (e.g., HTML, JSON, XML). However, there will be situations when the resource needs to be represented by binary media. Here are a few possible use cases:

- A customer wants to download a PDF version of an invoice.
- A traveler boarding a plane needs to display a scannable image of a bar code.
- A worker wants to watch a video of assembly instructions.

The SRP HTTP Framework includes a sample Picture API. It includes logic that returns a head shot graphics for the sample Contacts resource. Reviewing this code (found in the `PICTURE_API` routine) will be useful to see how a fully integrated image API can be written, we'll summarize below the four basic steps for **returning binary media in your API**.

## Step One: Read the Binary Data

Your binary data needs to be read into a variable. Some applications store binary media inside of an OpenInSight database so a [Read](#) statement might be used. The Picture API assumes its images are stored in a specific OS folder, so it uses an [OSRead](#) statement instead. Here is the relevant code from the Picture API:

```
OSRead PictureBinary from PicturePath then
    // Other logic goes here.
end
```

## Step Two: Set the Response Body

As with text based media, use the [SetResponseBody](#) service to identify the content to be returned to the caller:

```
HTTP_Services('SetResponseBody', PictureBinary)
```

## Step Three: Identify the Response as Binary

The above *SetResponseBody* example uses proper syntax and would work adequately for most text based media, but binary media needs to be packaged differently for the OECGI to return it correctly. This is done by setting the *IsBinary* argument to a 1:

```
Equ True$ to 1
HTTP_Services('SetResponseBody', PictureBinary, True$)
```

## Step Four: Set the Correct Content-Type

Although our response will now be properly packaged up as binary, the client will likely not be able to process the media properly. Should it attempt to display this as a PDF document? Should it attempt to play this as a video? Clients normally need this additional metadata to know how to proceed. Therefore, the *Content-Type* response header needs to be set with the appropriate [IANA media type](#). There are two ways of doing this. One way is to use the [SetResponseHeaderField](#) service and type in the field name (i.e., *Content-Type*) and field value (e.g., *image/jpeg*). The easier way is to use the *Content Type* argument that has been provided with the *SetResponseBody* service. The following updated code sample uses this latter method:

```
Equ True$ to 1
HTTP_Services('SetResponseBody', PictureBinary, True$, 'image/jpeg')
```