

# Handling Blobs in the REST Interface

The Designer can handle blob parameters in the body of a REST interface for requests and responses. Blobs may contain different content types. How the content types should be handled can be configured on client and server side by four attributes provided by stereotype **REST Operation** for REST APIs and REST adapters.



For REST adapters, you can change the options listed below on manually created REST interfaces, or once you have copied your connector definitions to the **API** folder of your service.

Find below an overview on these attributes and what combinations make sense.

Direction	Attribute	Description	Adapter Implementation	API Implementation
Request	<b>Accepted Request Content Type</b>	Defines the content types of the request body.	Specify which content type you are sending with your request. The Runtime generates a matching "Content-Type" header to your request.	Define which content types the endpoint accepts.
	<b>Reject Other Request Content Types</b>	Triggers a content type check by the xUML Runtime.	<p>Specifies if the REST adapter should reject calls to this endpoint if a request uses a custom content type that does not match the content types listed in <b>Accepted Request Content Type</b>.</p> <p>This can be the case if you set the "Content-Type" header manually. If set to true, and the "Content-Type" header does not match, the adapter does not contact the endpoint at all but rejects calls with an exception (Endpoint '&lt;endpoint URL&gt;' does not declare '&lt;content type&gt;' as accepted content type).</p>	<p>Define if other content types than defined with <b>Accepted Request Content Type</b> should be rejected with HTTP 415 (Unsupported Media Type).</p> <p>The Runtime reads the related "Content-Type" header and compares it to the content types that are accepted by the service (<b>Accepted Request Content Type</b>).</p>
Response	<b>Blob Body Content Type</b>	Defines the content type of the response body.	Specify a list of content types you accept as a response. The Runtime generates a matching "Accept" header to your request.	Define which content types the endpoint can return to the client.
	<b>Reject Other Response Content Type</b>	Triggers a content type check by the xUML Runtime.	<p>Specifies if the REST adapter should reject calls to this endpoint if a request uses a custom content type that does not match the content types listed in <b>Blob Body Content Type</b>.</p> <p>This can be the case if you set the "Accept" header manually. If set to true, and the "Accept" header does not match the content types the service can provide, the adapter does not contact the endpoint at all but rejects calls with an exception (Set "accept" header does not accept any of declared response content types).</p>	<p>Define if the service should reject the request with HTTP 406 (Unsupported Media Type) if the client request has accept headers that differ from the ones defined with <b>Blob Body Content Type</b>.</p> <p>The Runtime reads the related "Accept" header and compares it to the content types that can be provided by the service (<b>Blob Body Content Type</b>).</p>

## On this Page:

- [Examples](#)
  - [Good Cases](#)
  - [Error Cases](#)

## RESTAPI\_BlobContent\_Example



Click the icon to download a simple example model that shows the implementation of a REST API that receives and returns Blob content with **Scheer PAS Designer**.

## RESTAdapter\_BlobContent\_Example



Click the icon to download a simple example model that shows how to handle Blob content with a REST call in **Scheer PAS Designer**.

## Related Pages:

- [Manually Providing the REST Interface](#)
- [Handling Blobs in the REST Interface](#)
- [Setting REST Request Options](#)
- [Getting the REST Adapter Response](#)
- [REST Adapter Error Handling](#)
- [REST Adapter Reference](#)

## Examples

The **RESTAPI\_BlobContent\_Example** and the **RESTAdapter\_BlobContent\_Example** implement a server and a client that exchange blob content (pictures). They have the following blob content setup implemented:

Server (RESTAPI_BlobContent_Example)		Content Type Settings		Description
Store a picture	POST /picture	<b>Accepted Request Content Type</b>	image/png; image/jpeg	The server can store PNGs and JPGs.
Return a picture by id	GET /picture/<an id>	<b>Blob Body Content Type</b>	image/png	The server returns PNGs upon request.

Client (RESTAdapter_BlobContent_Example)		Content Type Settings		
Send a picture	POST /picture	Accepted Request Content Type	image/png	The client sends PNGs.
Retrieve a picture	GET /picture/<an id>	Blob Body Content Type	image/png; image/jpeg	The client accepts PNGs and JPGs as a response.

## Good Cases

The following table shows some example calls and how they would be executed.

Client			Generated Header	Server
1	Store a picture	POST /picture	Content-Type: image/png	accepts request
				returns id HTTP 201
2	Retrieve a picture	GET /picture/123456	Accept: image/png	accepts request
			Content-Type: image/png	returns picture PNG

## Error Cases

The following table shows some error cases and their consequences if you do not rely on the automatically generated headers but set your own headers instead.

Example	Client	Custom Header	Server
1 Store a picture, wrong content type	POST /picture	Content-Type: image /svg+xml	accepts request
			<ul style="list-style-type: none"> <li>handles wrong content type internally</li> <li>RESTLM/11: Unknown input format (image /svg+xml)</li> </ul>
2 Store a picture, wrong content type, server rejects other content types	POST /picture	Content-Type: image /svg+xml	Reject Other Request Content Types = <b>true</b>
			<ul style="list-style-type: none"> <li>RESTLM/11 Unknown input format (image /svg+xml)</li> <li><b>HTTP 415</b> (Unsupported Media Type)</li> </ul>
3 Store a picture, wrong content type, client rejects other content types	POST /picture Reject Other Request Content Types = <b>true</b>	Content-Type: image /svg+xml	not called at all
	<ul style="list-style-type: none"> <li>RESTLM/7: Endpoint '&lt;endpoint URL&gt;' does not declare '&lt;content type&gt;' as accepted content type</li> </ul>		
4 Retrieve a picture, wrong content type	GET /picture/123456	Accept: image /svg+xml	accepts request

				<ul style="list-style-type: none"> <li>• handles wrong content type internally</li> <li>• RESTLM/10: Cannot generate any of the expected output formats (image/svg+xml)</li> </ul>
5	Retrieve a picture, wrong content type, server rejects other content types	GET /picture/123456	Accept: image/svg+xml	<b>Reject Other Response Content Type = true</b>
				<ul style="list-style-type: none"> <li>• RESTLM/10: Cannot generate any of the expected output formats (image/svg+xml)</li> <li>• <b>HTTP 406</b> (Not Acceptable)</li> </ul>