


REST Adapter Reference

Stereotype Attributes

REST Alias

Attribute (Name of Setting)	Description	Allowed Values / Examples	
Name	Specify a name for the alias.	any string	
Standard			
Protocol (protocol)	Specify here the protocol through which the REST service is accessible.	http, https	
Port (port)	Specify here the port through which the REST service is accessible.	a valid port	
Host (host)	Specify here the host running the REST service.	a valid host	
Base Path (basePath)	<div><div> Please note that the base path differs whether you want to access a Designer service from the outside of PAS or from the inside. When you trigger Designer services within the platform, use the part of the path starting with <code>/rest/....</code></div></div> <div>Specify here the base path of the REST service.</div>	<div>a valid path, e.g</div> <ul style="list-style-type: none"><code>/Idea_Management/rest/process/Idea_management</code> if you want to access a Designer service from outside PAS<code>/rest/process/Idea_management</code> if you want to access a Designer service from inside PAS	
Ignore HTTP Errors (ignoreHttpErrors)	Specify here whether you want the REST adapter to throw an exception upon receiving an HTTP error code <code>>= 400</code> .	true	Do not throw an exception upon receiving an HTTP error code <code>>= 400</code> (default).
		false	Throw an exception upon receiving an HTTP error code <code>>= 400</code> .
Additional Request Headers (additionalHeaders)	This attribute can contain a list of additional headers in form of name/value pairs.	Valid format is: <code><name>: <value></code> , e.g. <code>API-Key:acme</code> . Separate multiple headers with a comma.	
Advanced			
HTTP Header Role Definitions for requests (requestHttpHeaderRoles)	In the context of HTTP based adapters (URL, REST , SOAP), enable automatic header generation for the listed headers. These definitions overwrite the default behavior, and X-Transaction-Id , X-Request-Id , X-Sender-Host and/or X-Sender-Service will be substituted by this definition. requestHttpHeaderRoles can hold a list of definitions in format <code><http header name>:<role></code> , that will automatically be generated for each adapter call on this alias. <code><role></code> can be one of the listed allowed values (one list entry per line).	client_host	Provide the client host in a header <code><http header name></code> instead of X-Sender-Host .
		client_service	Provide the client service in a header <code><http header name></code> instead of X-Sender-Service .
		correlation_id	Provide the correlation ID in a header <code><http header name></code> instead of X-Request-Id .
		transaction_id	Provide the transaction ID in a header <code><http header name></code> instead of X-Transaction-Id .
		passthrough	Pass a present header <code><http header name></code> to the called service.

On this Page:


- [Stereotype Attributes](#)
 - [REST Alias](#)
 - [REST Adapter](#)
 - [REST API](#)
 - [REST Operation](#)
 - [REST Parameter](#)
 - [REST Error \(Class\)](#)
 - [REST Port \(Class\)](#)
 - [REST Resource \(Class\)](#)
 - [REST Error Code \(Property\)](#)
 - [REST Error Message \(Property\)](#)
- [REST Adapter Parameters](#)
- [REST Utility Functions](#)
- [REST Content Types](#)
- [REST Parameter Types](#)
 - [AdapterResponse](#)
 - [Request](#)
 - [RequestOptions](#)
 - [Response](#)
 - [Request and Response Types](#)

RESTAdapter_SupportManager_Example




Click the icon to download a simple example model that shows the usage of the REST adapter in **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**.

RESTAdapter_CallProcessWithRoles_Example



Click the icon to download a simple example model that shows how to call a subprocess from a parent process with a REST call in **Scheer PAS Designer** with special considerations of role handling.


		passth rough =<req uest head er name>	Pass an present header <request header name> to the called service under the name of <http header name>. This is equivalent to renaming a header.
Digest Algorithm (digestAlgor ithm)	Generates a HTTP digest header using the specified algorithm. When applied, a digest header is generated using the specified algorithm, and sent with the request. The generated header conforms with RFC3230 and RFC5843. <div> Only one value is supported (no multi-value header).</div>	None	No header generated (default).
		MD5	Generate header using MD5 algorithm.
		SHA	Generate header using SHA algorithm.
		SHA-1	Generate header using SHA-1 algorithm.
		SHA- 256	Generate header using SHA-256 algorithm.
		SHA- 512	Generate header using SHA-512 algorithm.
Accepted Count of Redirects to Follow (followRedir ects)	Specify here the maximum number of redirects to follow. Default value is 0 (no redirects).	any integer, 0 (default)	
cURL O pti ons (options)	Specify native cURL options as listed in Setting cURL Options on the URL Adapter . Use one of the following syntax rules: <ul style="list-style-type: none">values separated by ' , ' in one linevalues separated by ' ' in one linelist of attributes		
JSON: Keep Null Values (jsonKeepN ulls)	When jsonKeepNulls is true, attributes of the REST parameter having NULL values will be provided with the REST call, otherwise they will be left out completely (see also chapter NULL Values).	true	Render attributes with NULL values to the REST call.
		false	Leave out attributes with NULL values in the REST call (default).
JSON: Compact Format (jsonCompa ct)	When jsonCompact is true, the JSON composer will generate compact JSON, otherwise it will generate pretty JSON.	true	Generate compact JSON (default).
		false	Generate pretty JSON.
JSON: Write Type Discriminat or		true	
		false	
Authentication			
User (user)	Specify credentials here if the called REST service needs basic authentication.	Valid format is <user> /<password>, e.g. david. stringer/pAsWoRd456	
Proxy Settings (if the called REST service is accessed via a proxy)			
Proxy: URL (proxyURL)	Specify the URL of the proxy server.	See CURLOPT_PROXY .	
Proxy: User (proxyUser)	Specify the proxy credentials.	See CURLOPT_PROXYUSERP WD , format is <user> /<password>, e.g. david. stringer/pAsWoRd456	
Proxy: Type (proxyType)	Specify the proxy type.	See CURLOPT_PROXYTYPE .	
SSL Settings (if the called REST service uses SSL)			


Related Pages:

- [REST Adapter](#)

SSL: CA Info (sslCAInfo)	Specify a file name containing additional certificates for the connection verification (e.g. additional root CAs).	See CURLOPT_CAINFO .
SSL: Certificate File (sslCertificateFile)	Specify a file name containing the client certificate.	See CURLOPT_SSLCERT .
SSL: Certificate Type (sslCertificateType)	Specify the type of the certificate.	See CURLOPT_SSLCERTTYPE .
SSL: Private Key File (sslPrivateKeyFile)	Specify a file name containing the private key.	See CURLOPT_SSLKEY .
SSL: Private Key Password (sslPrivateKeyPassword)	Specify the password for the private key.	See CURLOPT_KEYPASSWD .
SSL: Private Key Type (sslPrivateKeyType)	Specify the type of the key.	See CURLOPT_SSLKEYTYPE .
SSL: Verify Host (sslVerifyHost)	Specify whether to verify the host information from the SSL connection.	See CURLOPT_SSL_VERIFYHOST .
SSL: Verify Peer (sslVerifyPeer)	Specify whether to verify the peer information from the SSL connection.	See CURLOPT_SSL_VERIFYPEER .

REST Adapter

Attribute (Name of Setting)	Description	Allowed Values / Examples
Alias (alias)	Specify the REST alias the adapter should use to establish the connection.	a valid REST alias
Digest Algorithm (digestAlgorithm)	Generates a HTTP digest header using the specified algorithm. When applied, a digest header is generated using the specified algorithm, and sent with the request. The generated header conforms with RFC3230 and RFC5843. <div>  Only one value is supported (no multi-value header). </div>	No No header generated (default).
		MD5 Generate header using MD5 algorithm.
		SHA Generate header using SHA algorithm.
		SHA-1 Generate header using SHA-1 algorithm.
		SHA-256 Generate header using SHA-256 algorithm.

Reject Other Response Content Types (rejectOtherResponseContentType)	The xUML Runtime performs a verification of the content-type header for REST calls. Specify whether to return an error for requests with a content type that does not conform with the content types specified in Blob Body Content Type . Refer to Handling Blobs in the REST Interface for a deeper explanation and some examples.	true	<ul style="list-style-type: none"> Reject to perform adapter call if the "Content-Type" header does not match the value listed in Blob Body Content Type (default). Exception: RESTLM/48: Request content type not declared as accepted by the service
		false	Perform the adapter call in spite of "Content-Type" header mismatch, and let the service handle this.
Accepted Request Content Types (acceptedRequestContentType)	Provide a list of content types you accept as a response. This must be a list of valid "accept" headers as defined in RFC 7231 . The Runtime generates a matching "Accept" header to your request. Refer to Handling Blobs in the REST Interface for a deeper explanation and some examples.	a list of valid media ranges	e.g. <code>image/jpeg;image/png</code> Default is <code>application/octet-stream</code> if not specified.
<div>  This attribute must be left unset if no Blob output parameters are used. In future versions, the effect of this attribute may be extended to other contexts as well. </div>			
Reject Other Request Content Types (rejectOtherRequestContentType)	Specify whether to return an error for responses with a content type that does not conform with the content types specified in Accepted Request Content Type . Refer to Handling Blobs in the REST Interface for a deeper explanation and some examples.	true	<ul style="list-style-type: none"> Reject to perform adapter call if the content types the service returns (Blob Body Content Type of service) does not match the values listed in Accepted Request Content Type (default). Exception: Set "accept" header does not accept any of declared response content types
		false	Perform the adapter call in spite of "Accept" header mismatch and let the service handle this.

REST Parameter

Tagged Value	Description	Allowed Values		Allowed REST Methods	Allowed Types	Hints and Limitations
In (in)	Defines how the parameter will be passed to the REST method. This tag is mandatory .	query	via a query string	all	all simple types and Array of simple type	Unknown parameters will be ignored, known will be passed to the method after being URL-decoded.
		path	via the REST resource path	all	Integer, Float, String, Boolean, DateTime	Path parameters are all required. All path parameters must be consumed by the called method and the parameter names must be the same as the path segment identifiers (without colon).
		body	via the REST call body	POST, PUT, PATCH	a complex type and Array	A REST method can have only one body parameter.
		header	via the REST call header	all	all simple types and Array of simple type	Unknown parameters will be ignored, known will be passed to the method.

External Name (externalName)	Defines an external name for the REST parameter	any string	Use this, when wanting to access a REST service that has parameter names with special characters. In this case, set this name (e.g. ugly@parameter-name) to externalName and give a better name. So you will not have to escape the parameter every time you use it.
--	---	------------	---

REST Error (Class)

The stereotype **REST Error** is used on a class to mark it as REST error class. Each REST port has a error class assigned that, in case of error, contains additional information. You can get this information using something like

```
create error;
set error = adapterResponse.responseObject;
```

if error is an object of the type indicated by this stereotype.

REST Port (Class)

Attribute	Description	Allowed Values / Examples	
path	Defines the path to this rest interface. If empty, the path is derived from the package structure.	none	path of the package structure will be used, e.g. /Services/SupportCase/SupportAPI
		any valid path string	path string starting with "/", e.g. /support
errorClass	Assigns a user-defined REST Error class to the REST interface. This class should be set in case of error and given back via the REST response.	any complex type describing the structure of the error	
apiVersion	Defines the API version this port type provides (for documentation purposes only).	any string, default is 1.0.0	

REST Resource (Class)

Attribute (Name of Setting)	Description	Allowed Values	
Relative Path (relativePath)	Defines the path of the REST resource or collection in relation to the parent resource. You can provide a static path, or a dynamic path using the notation :<name of a REST Parameter>. You may also provide a combination of both.	none	the name of the REST resource will be used, e.g. /supportcases
		any valid string	the given name will be used
		a dynamic path supplying a REST parameter	dynamic path, the value of the REST parameter will be passed to the REST methods, e.g. :id
Is Verbatim Path (isVerbatimPath)	Disable most of the path normalization. All escaping must be done manually, leading or trailing whitespaces are preserved. This is what still will be done, if isVerbatimPath is set to true: <ul style="list-style-type: none"> variable discovery The xUML Runtime will try to find and replace variables (:<my variable name>). colon escaping If the path contains a colon that (and that does not indicate a path parameter), you can escape this colon by another one, like :: (e.g. /myAPI/resource/aText::anotherText/:variable). correction of segment delimiters The xUML Compiler adds a leading slash (if not present), removes trailing slashes (if present) and removes duplicates slashes, e.g. myAPI/resource/:variable/ becomes /myAPI/resource/:variable. 	true	Path should be treated as verbatim, path normalization is disabled.
		false (default)	Path should be URL encoded.

REST Error Code (Property)

REST Error Code and **REST Error Message** are applied to one property of the **REST Error** class each to define that the corresponding properties contain the error code and the error message of the REST error. You cannot set **REST Error Code** on two properties of the same error class.

REST Error Message (Property)

REST Error Code and **REST Error Message** are applied to one property of the **REST Error** class each to define that the corresponding properties contain the error code and the error message of the REST error. You cannot set **REST Error Message** on two properties of the same error class.

REST Adapter Parameters

Name	Type	Direction	Description
requestOptions	Request Options	in	Use this parameter to configure the REST Adapter dynamically and overwrite the settings from the component diagram.
response	Any	out	This parameter holds the adapter output and is of that type that is given back by the called REST service.

REST Utility Functions

Access to HTTP request and response objects is provided through global methods: `getRestHttpRequest()` and `getRestHttpResponse()`.

Function	Parameter	Return Value	Description	Example
getRestHttpRequest()	none	object of type Request	Returns the request details as provided by the HTTP call. Changing the request object will not have any effects. For details see getRestHttpRequest() .	<pre>local request = getRestHttpRequest();</pre>
getRestHttpResponse()	none	object of type Response	Set the response details to return them to the caller. For details see getRestHttpResponse() .	<pre>local response = getRestHttpResponse();</pre>

REST Content Types

With the REST Adapter, the xUML Runtime can handle JSON or XML as content types. The Runtime will parse either response content (JSON or XML) to a response object automatically. You do not need to set any headers. If no headers are set, the Runtime will use JSON as a default format and set **Accept** header to `application/json,text/json,application/xml;q=0.9,text/xml;q=0.9,*/*;q=0.8`.


Regarding response parsing, the xUML Runtime will process REST responses as follows:

- If the **Content-Type** header of the response is JSON or XML, respective format will be used (regardless what is specified in the **Accept** header).
- If the Content-Type header of the response is not set, the Runtime will assume that the content matches the **Accept** header of the request. If no **Accept** header has been specified, JSON is the fallback.
- In case of unsupported **Content-Type**, JSON is the fallback.

REST Parameter Types

AdapterResponse

Attribute	Type	Description	Allowed Values / Examples
httpStatus	Integer	HTTP status code of the adapter call.	500

headers	Array of HeaderField	<p>DeprecatedThis attribute is deprecated. It is only present for compatibility reasons.</p> <p>Please use HTTPHeaderMap (see below) for new implementations. The implementation of HTTPHeaderMap complies to the HTTP specification and supports multiple headers having the same name.</p>	
body	Blob	<p>HTTP body of HTTP response. Among others, the body contains</p> <ul style="list-style-type: none"> the response Object for successful calls an object of the RESError class if the call has not been successful. 	
responseObject	Any	<p>Response Object of the REST adapter call if the endpoint returns one (response output parameter).</p> <div>  The response object is only returned for successful calls. </div>	
HTTPHeaderMap	Map of Entry	<p>Header information as a map. The map contains arrays of header value strings whereas the header name is the key of the map.</p> <ul style="list-style-type: none"> Header names are lowercase and treated case insensitive. Multiple headers with the same name are treated as arrays. <p>Refer to HTTP Header Support for more information on the standard xUML HTTP headers.</p>	

Request

Attribute	Type	Description	Allowed Values / Examples
queryString	String	Query string, if provided with the call.	<code>status=in%20progress</code>
queryParameters	Array of Parameter	All query parameters as an array of Parameter classes containing name/value pairs.	
body	Blob	Body of the HTTP request.	
path	String	Path to the REST resource.	<code>/support</code> <code>/supportcases/</code>
pathParameters	Map	All REST parameters as a map.	
HTTPHeaderMap	Map of Entry	<p>Header information as a map. The map contains arrays of header value strings whereas the header name is the key of the map.</p> <ul style="list-style-type: none"> Header names are lowercase and treated case insensitive. Multiple headers with the same name are treated as arrays. <p>Refer to HTTP Header Support for more information on the standard xUML HTTP headers.</p>	

RequestOptions

Attribute	Type	Description	Allowed Values / Examples
additionalHeaders	Array of HeaderField	All REST request header fields as an array of HeaderField classes containing name/value pairs.	
options	Array of Option	<p>Specify native cURL options as listed in Setting cURL Options on the URL Adapter.</p> <p>Use one of the following syntax rules:</p> <ul style="list-style-type: none"> values separated by ' , ' in one line values separated by ' ' in one line list of tagged values 	

ssl	SSL	Use this parameter to supply SSL information.	
proxy	Proxy	Use this parameter to supply necessary proxy information.	
additionalQueryParameters	Array of Parameter	Use this parameter to provide additional query parameters to the REST service call.	
followRedirects	Integer	Specify here the maximum number of redirects to follow.	any integer
basicAuth	Authentication	This parameter provides an object of type Authentication containing the user and the password.	
basePath	String	Overwrite here the base path of the REST service.	a valid path, e.g. /support
host	String	Overwrite here the host running the REST service that has been defined in the component diagram.	a valid host, e.g. scheer-acme.com
port	Integer	Overwrite here the port through which the REST service is accessible.	a valid port number, e.g. 12345
protocol	String	Overwrite here the protocol through which the REST service is accessible.	http Use the HTTP protocol to access the service (default).
			https Use the HTTPS protocol to access the service.
ignoreHttpErrors	Boolean	If true, Default value is true.	true HTTP error codes ≥ 400 will not cause an exception in the service (default). This implies, that the response body is accessible even if HTTP errors occur.
			false HTTP error codes ≥ 400 will cause an exception in the service.
jsonComposerOptions	ComposerOptions	Use this parameter to specify JSON composer options on the REST call. You can use these options to e.g. overwrite jsonKeepNulls from the REST alias.	Valid composer options as described below .

Response

Attribute	Type	Description	Allowed Values / Examples
headers	Array of HeaderField	All HTTP response header fields as an array of HeaderField classes containing name/value pairs.	
statusCode	Integer	The resulting HTTP status code. If not set explicitly using this object, the service returns 200 if no exception occurred, or 500 otherwise.	404
responseObject	Any	Response Object of the REST adapter call if the endpoint returns one (response output parameter).	

Request and Response Types

REST Type	Attribute	Type	Description	Allowed Values / Examples	
Authentication	username	String	Specify a username.		
	password	String	Specify a password.		
Certificate	file	String	Provide name of a certificate file.		
	type	String	Specify the type of the certificate file.	One of PEM or DER.	
ComposerOptions	jsonKeepNulls	Boolean	Specify the handling of NULL values.	true	Keep NULL values during JSON composing.

				false	Do not serialize <code>NULL</code> values during JSON composing.
	jsonCompact	Boolean	Specify the generated JSON format.	true	Serialize to JSON without pretty-printing.
				false	Produce a pretty JSON.
	jsonWriteTypeDiscriminator	Boolean	Specify whether to write type information to the generated JSON.	true	Write a type discriminator to the generated JSON.
				false	Do not write a type discriminator to the generated JSON.
Entry	key	String	Specify the key of a map entry.		
	value	Array of Any	Provide a list of values of the map entry. The dynamic type for <code>HTTPHeaderMap</code> is String .		
HeaderField	name	String	Specify the name of the header field.		
	value	String	Provide a header value.		
Key	file	String	Specify the name of the file that contains the private key.		
	type	String	Specify the type of the private key.		
	password	String	Specify the password for the private key.		
Option	name	String	Specify the name of the cURL option.		
	value	String	Provide a value.		
Parameter	name	String	Specify the name of the parameter.		
	value	String	Provide a value.		
Proxy	url	String	Specify a URL to the proxy server.		A valid URL.
	type	String	Specify a proxy type	HTTP	The proxy is a HTTP proxy.
				SOCKS5	The proxy is a SOCKS5 proxy.
	authentication	Authentication	See above.		
SSL	verifyPeer	String	Specify whether to verify the peer information from the SSL connection.	true	Verify the peer information (default).
				false	Do not verify the peer information.
	verifyHost	String	Specify whether to verify the host information from the SSL connection.	On	Verify the host information (default).
				Off	Do not verify the host information.
				Existence	Only check for existence of the host.
	caInfo	String	Specify the name of the file that contains additional certificates for the connection verification.		
	certificate	Certificate	Specify the certificate information (see above).		
	key	Key	Specify the private key information (see above).		