

REST Adapter Error Handling

REST services in general return errors via the HTTP status code. Besides the HTTP status code there is no standard way of how REST services provide additional error information. Developers can return additional information in HTTP headers or body, though.

Wanting to call a REST service, you need to know how this service handles errors. With the xUML Runtime REST implementation, we decided to provide error information via an error class in the HTTP body.

REST adapter error handling involved the following steps:

1. [Catching the error after the REST adapter call.](#)
2. [Reading the error object from the adapter response.](#)

Catching the Error

Catching REST adapter errors depends on the flag **Ignore HTTP Errors** on the alias (see below for more details).

You can override this flag in your REST request via the request options (see [Setting REST Request Options](#)).

Ignore HTTP Errors = true

With tagged value **Ignore HTTP errors** set to true, the xUML Runtime will not throw any exception if a HTTP status code ≥ 400 is returned by the REST service. You should read the **HttpStatus** from the adapter response (see [Getting the REST Adapter Response](#)) and decide yourself if any additional error treatment is needed.

Ignore HTTP Errors = false

With tagged value **Ignore HTTP errors** set to false, the xUML Runtime will throw an exception for HTTP status codes ≥ 400 . You can catch these exceptions directly after a REST Adapter call as described on [Catching Errors](#) and do some error handling.

To get more details on the error, you can:

- **call [getError\(\)](#)**
This will return the Runtime error details such as error code, description, call stack, timestamp and so on.
- **read the REST Adapter response**
In case of error, the REST service may provide an error object as a response. You can read this error object from the adapter response (see [Getting the REST Adapter Response](#)).

Reading the Error Object From the Adapter Response

Look at the activity diagram below. It shows the implementation of closing all selected support cases. If a support case cannot be closed for whatever reason, an error message is logged into the service log.

On this Page:

- [Catching the Error](#)
 - [Ignore HTTP Errors = true](#)
 - [Ignore HTTP Errors = false](#)
- [Reading the Error Object From the Adapter Response](#)

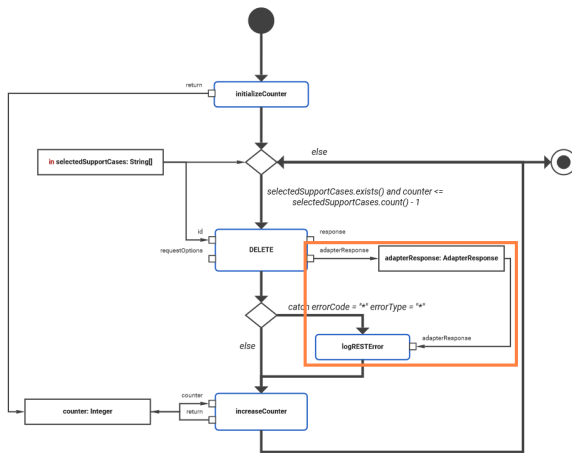
RESTAdapter_SupportManager_ Example



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

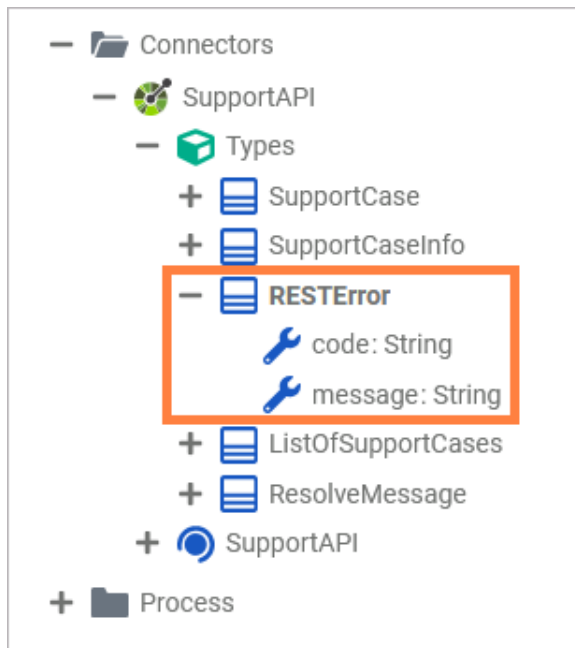
Related Pages:

- [Setting REST Request Options](#)
- [Getting the REST Adapter Response](#)
- [Catching Errors](#)

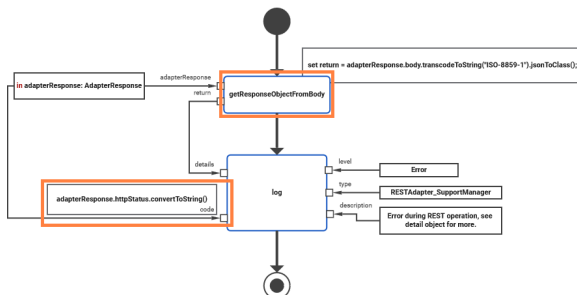


As described on [Getting the REST Adapter Response](#), you can get the adapter response directly from the adapter call. The REST Adapter response contains the HTTP status, HTTP headers, a REST body and a REST response object **responseObject**. In case of error and if a Designer service was called, the REST body contains an error detail object.

The type definition of this error object has been imported with importing the OpenAPI file of the REST service and can be found in the imported module. In this example, this is **RESError**.



You can evaluate the error details to e.g. log them to the service log as displayed below.



This implementation is explained in detail on [Getting the REST Adapter Response](#).