REST Adapter Error Handling

This page explains the **REST Adapter** in Bridge context. If you were looking for the same information regarding the PAS Designer, refer to REST Adapter in the Designer guide.

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 E2E Bridge REST implementation, we decided to provide error information via an error class in the HTTP body.

On this Page:

Related Pages:

- Getting the REST Adapter Response
- Catching Errors

Catching Errors

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

You can override this flag per request via the request options (see Setting REST Request Options).

Ignore HTTP Errors = true

Bridge 6.0.60 Builder 6.0.22With tagged value **Ignore HTTP errors** set to true, no exception will be thrown upon HTTP status codes >= 400 given back by the REST service. You should read the **httpStatus** from the adapter response (as described on 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 E2E 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).

This also applies to older models (before Builder 6.0.22) that do not have this flag present.

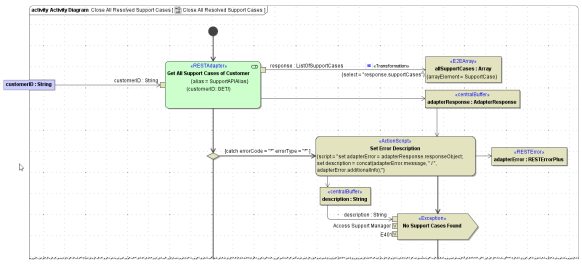
Reading the Error Object From the Adapter Response

Example File (Builder project Add-ons/REST):

<your example path>\Add-ons\REST\uml\accessSupportManager.xml

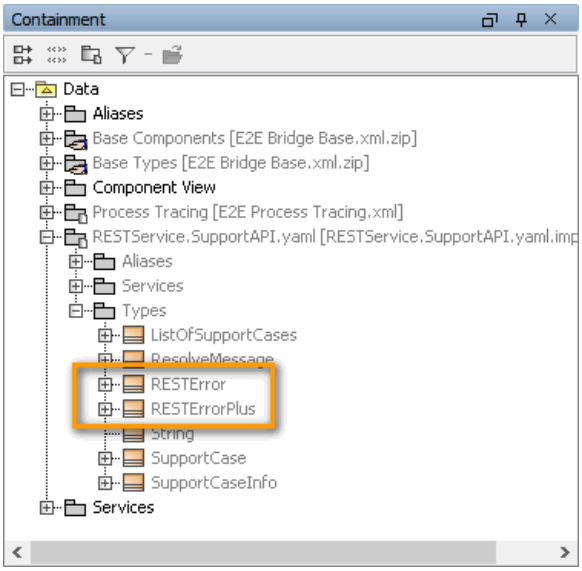
Look at the activity diagram below. To close all support cases of a specific customer, you have to get all support cases first. This is done by an adapter call (**GET /**) to the REST_SupportManagerExample service.

Figure: Reading the Adapter Response To An Error Class

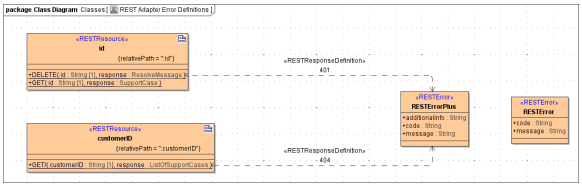


As described on Getting the REST Adapter Response, you can get the adapter response directly from the adapter call. The REST Adapter response contains the HTTP status, HTTP headers, a REST body and a **REST response object**. In case of error, the response object contains a REST error object provided by the REST service.

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



Depending on the HTTP status code, the service may return two different error objects. Refer to Defining a REST Service Interface for more information on the relation between error classes and HTTP status codes.
To get a better overview on the error definitions, you can drag all REST error classes and REST resources to a class diagram and select **Related Elements > Display Paths** from the context menu.



As you can see from here, the operation **GET/** we are using to get all support cases of a specific customer returns **RESTErrorPlus** in case of HTTP status 404 (not found).

Get the response object from the adapter response to the appropriate error class with `set adapterError = adapterResponse.responseObject;`. **adapterError** should in this case be of type **RESTErrorPlus**.