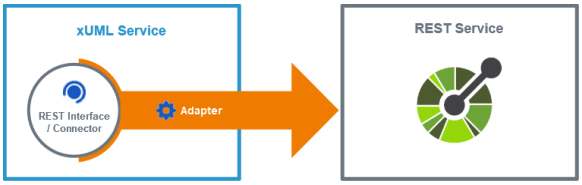
REST Adapter

REST (Representational State Transfer) is a way to provide access to web resources using a uniform and predefined set of stateless operations. Refer to the Wikipedia pages of Representational State Transfer for more information on REST.

Today, there are already many services, which speak REST. The Designer has a REST adapter ready. It enables you to use any REST service as a backend for your service implementations.



A REST adapter needs information on the REST API to be called, means the REST resources and their interface. You can provide this information in two ways:

- **Via a connector** (see image above)
  This is the easiest way to provide the necessary information. The Designer allows for importing different REST API configurations (see Creating an OpenAPI Connector for more information about that).
- **Via a manual data model**
  If you do not have a configuration file to import at hand, you can create the necessary resources and interface definitions manually to the **Implementation** folder (see Manually Providing the REST Interface for more).

OpenAPI Connector for REST Services

Each Web service has its own distinct interface - in case of a REST service, defined by the names of the resources, their operations and their parameters. Before you are able access an external web service from your service model, its interface definition must be imported to the service model.
With the Designer, you can create a connector based on OpenAPI 2.0 Specification service descriptors encoded in YAML (Swagger). The connector contains

- the **REST interface** that describes how resources of this endpoint can be accessed,
- a **REST alias** that specifies where the REST service is located and more connection options.

For more details on connectors, refer to

- Creating an OpenAPI Connector for more information on how to create a connector by importing a descriptor.
- Using Connectors for more information on how to use the elements provided by the connector in your service model.
- Aliases for more information on how to configure an alias.

Accessing a REST Resource

Once you have created a connector to your service (or created the necessary data model by hand), you can add REST operations from the connector to any diagram to create a REST adapter to your service.

1. Add an operation from the REST connector to a diagram to add a REST adapter.
   You can drag out operations from the data model to any diagram:
   - BPMN execution diagram
   - mapping diagrams
   - activity diagrams
2. Configure the REST adapter to your needs.
   REST adapter operations need to get stereotype **REST Adapter** applied. Refer to REST Adapter Reference for more information on the configuration options of this adapter.

Refer to Using Connectors for more.

On this Page:

- OpenAPI Connector for REST Services
- Accessing a REST Resource

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.

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

- Creating an OpenAPI Connector
- Using Connectors
- Aliases

Related Documentation:

- Representational State Transfer
- OpenAPI 2.0 Specification