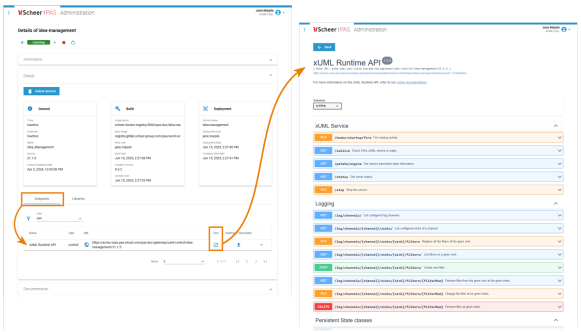
xUML Service Interface

The service state machine and the process (sub) state machine as described on xUML Service State Machines are accessible via REST interfaces. You can use these interfaces to get information on the service's/process' states, and to trigger state transitions.

Using the xUML Runtime Interface

The PAS platform features a n xUML Runtime API for each service. You can use this interface to obtain information on the states of the service's state machines in general, and to trigger state transitions. Access to this interface is available via the service details in the PAS Administration, section **Details >** tab **Endpoints**:

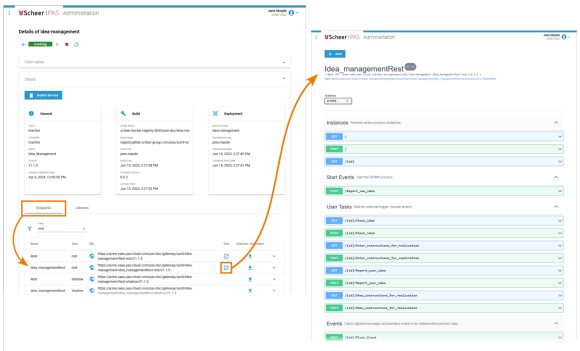


You can collect information on the persistent state classes, objects and pending events, or send signals like e.g. the **Abort** signal. The kind of information you can get, partly overlaps with the information you can get from the service interface (see below).

Refer to the reference page of the xUML Runtime API for a comprehensive list of all available requests.

Using the Service Interface

The BPMN process itself also has a dedicated REST API to gather information on dedicated process instances and their state(s), and to trigger transitions. Access to this interface is available via the service details in the PAS Administration, section **Details >** tab **Endpoints**:



As per default, the interface of a BPMN service has two GET requests:

- One to gather information on active processes: `GET /`
  This request returns a list of all ids of all active process instances.
- Another to get details of a dedicated process: `GET {id}`
  This request returns information on a dedicated process instance identified by their internal process id.

Via POST requests, you can send calls to the BPMN process itself.

- There is always one POST request per BPMN start event to start the process. This REST call triggers the creation of a process instance, means an instance of the root state machine described on xUML Service State Machines.
  In the example above, this would be `POST /Report_new_idea`.
- The other POST requests reflect process input, e.g. incoming messages from user tasks (forms), receive tasks, or events.

On this Page:

- Using the xUML Runtime Interface
- Using the Service Interface

Related Pages:

- xUML Service State Machines
- BPMN Process API Reference

Related Documentation:

- Administration Guide
  - Controlling Containerized xUML Services
  - xUML Runtime
    - xUML Runtime API
    - xUML Runtime API Reference
- Bridge Integration Platform
  - xUML Runtime API
  - xUML Service Details > REST Ports

In the example above, the POST request are related to forms that are displayed during process execution, e.g.

- `POST /{id}/Check_idea` sends the decision of the superior to the process.
- `POST /{id}/Enter_intructions_for_realization` allows the superior to provide further instructions.

Both example POST requests will trigger the process to continue when it is waiting for user input.

Go to BPMN Process API Reference for an overview on the details of the service API.