

# xUML Runtime API Reference

## On this Page:

- [xUML Service](#)
- [Logging](#)
- [Persistent State](#)
  - [Persistent State Classes](#)
  - [Persistent State Objects](#)
  - [Persistent State Objects of <class name>](#)
  - [Persistent State Owners](#)
  - [Persistent State Events](#)
- [Asynchronous Tracing](#)

## Related Pages:

- [Retrieving Persistent State Metadata with the PersistentStateControl Adapter](#)
- [Bridge API](#)

Bridge 7 Find below a list of resources that can be accessed via the xUML Runtime API. All resources refer to base path **/api**.

## xUML Service

Resource	Method	Description	API Version	Runtime Version
/isAlive	GET	Check if the xUML service is ready. A success HTTP code indicated readiness, an error HTTP code indicates startup still in progress.	1.2.0	Runtime 2021.6
/hooks/shutdown/fire	PUT	Fire the service shutdown activity.	1.1.0	Runtime 2020.13
/hooks/startup/fire	PUT	Fire the service startup activity.	1.1.0	Runtime 2020.13
/scheduler/events/{schedulerName}/fire	PUT	Fire the scheduler event (without checking the conditions).	1.1.0	Runtime 2020.13
/timer/events/{timerName}/fire	PUT	Fire the timer event (without checking the conditions).	1.1.0	Runtime 2020.13
/timer/events/{eventObserverActivity}/fire	PUT	Fire the event (without checking the conditions).	1.1.0	Runtime 2020.13
/pstate/engine	GET	Check if the xUML service uses persistent state and get general information on the persistent state configuration.	1.0.0	Runtime 2019.5
/status	GET	Get the status of the persistent state server.	1.0.0	Runtime 2019.5
/stop	PUT	Stop the running xUML service.	1.0.0	Runtime 2019.5

## Logging

Resource	Method	Description	API Version	Runtime Version
/log/channels/	GET	List all configured log channels.	1.1.0	Runtime 2020.1
/log/channels/{channel}/sinks	GET	List all configured sinks of a channel.	1.1.0	Runtime 2020.1
/log/channels/{channel}/sinks/{sink}/filters	PUT	Replace all filters of the given sink.	1.1.0	Runtime 2020.1
/log/channels/{channel}/sinks/{sink}/filters/	GET	List filters on a given sink.	1.1.0	Runtime 2020.1
/log/channels/{channel}/sinks/{sink}/filters/	POST	Create a new filter.	1.1.0	Runtime 2020.1

/log/channels/{channel}/sinks/{sink}/filters/{filterNum}	DELETE	Remove the filter with the given number.	1.1.0	Runtime 2020.1
/log/channels/{channel}/sinks/{sink}/filters/{filterNum}	GET	Retrieve the filter with the given number from the given sink.	1.1.0	Runtime 2020.1
/log/channels/{channel}/sinks/{sink}/filters/{filterNum}	PUT	Change the filter with the given number.	1.1.0	Runtime 2020.1

## Persistent State

The **Persistent State Control Adapter** gives access to persistent state metadata directly from within a service (self context). The same data can be retrieved using the **xUML Runtime API**.

If you want to retrieve metadata of persistent state of the very same service, always use the [Persistent State Control Adapter](#).  
If you want to retrieve data from other services, use the [xUML Runtime API](#).

## Persistent State Classes

Resource	Method	Description	API Version	Runtime Version
/pstate/classes/	GET	List all available persistent state classes.	1.0.0	Runtime 2019.5
/pstate/classes/{class}	GET	Get the persistent state object/state counters for a given class.	1.0.0	Runtime 2019.5
/pstate/classes/{class}/meta	GET	Get the persistent state metadata of a given class.	1.0.0	Runtime 2019.5

## Persistent State Objects

Resource	Method	Description	API Version	Runtime Version
/pstate/objects/{objectId}	GET	Get the metadata of a given persistent state object. The object is identified by its ID.	1.0.0	Runtime 2019.5
	DELETE	Delete a persistent state object by its ID.	1.0.0	Runtime 2019.5
/pstate/objects/{objectId}/events	GET	Get all pending events of a persistent state object. The object is identified by its ID.	1.0.0	Runtime 2019.5
/pstate/objects/{objectId}/signals/completion	POST	Send a completion signal to a persistent state object. The object is identified by its ID and the state it is stalled in.	1.0.0	Runtime 2019.5
/pstate/objects/{objectId}/states	GET	Get all states a persistent state object is in and possible transitions. The object is identified by its ID.	1.0.0	Runtime 2019.5
/pstate/queryObjects	POST	Globally, find persistent state objects matching the search criteria given by parameter.	1.0.0	Runtime 2019.5

## Persistent State Objects of <class name>

Resource	Method	Description	API Version	Runtime Version
/pstate/classes/<class name>/object	GET	Find a persistent state object of a given class by its primary key.	1.0.0	Runtime 2019.5
/pstate/classes/<class name>/objects/{objectId}	GET	Get the details of a persistent state object of a given class by its object ID.	1.0.0	Runtime 2019.5
/pstate/classes/<class name>/objects/{objectId}/signals/<signal name>	POST	Send the given signal to a persistent state object. The object is identified by its ID.	1.0.0	Runtime 2019.5
/pstate/classes/<class name>/queryObjects	POST	Find all persistent state objects matching the search criteria given by parameter.	1.0.0	Runtime 2019.5

## Persistent State Owners

Resource	Method	Description	API Version	Runtime Version
----------	--------	-------------	-------------	-----------------

/pstate /owners/	GET	Get a list of all persistent state owners and some details on the xUML services and objects they own.	1.0.0	Runtime 2019.5
---------------------	-----	---	-------	----------------

## Persistent State Events

Resource	Method	Description	API Version	Runtime Version
/pstate/events/{eventId}	GET	Get the details of a given persistent state event.	1.0.0	Runtime 2019.5

## Asynchronous Tracing

The API of asynchronous tracing allows to manage the trace settings and trace data of a service.

- A **classifier** in this context is a model element that can be traced.
- An **identifier** in this context identifies the recorded trace data.

You can clear all tracing of a service by performing

1. DELETE on /trace/configurations/ to remove all active traces.
2. DELETE on /trace/identifiers/ to remove all trace data.

Resource	Method	Description	API Version	Runtime Version
/trace/configurations/	GET	List all asynchronous trace configurations of this service.	1.1.0	Runtime 2020.14
/trace/configurations/	POST	Insert a new asynchronous trace configuration to this service.	1.1.0	Runtime 2020.14
/trace/configurations/	DELETE	Remove all asynchronous trace configurations for this service.	1.1.0	Runtime 2020.14
/trace/configurations/{classifier}	GET	List all asynchronous trace configurations for the given classifier.	1.1.0	Runtime 2020.14
/trace/configurations/{classifier}	PUT	Replace the asynchronous trace configuration of the given classifier.	1.1.0	Runtime 2020.14
/trace/configurations/{classifier}	DELETE	Remove all asynchronous trace configurations for the given classifier.	1.1.0	Runtime 2020.14
/trace/configurations/{classifier}/ /identifiers/	GET	List all available trace identifiers of the the given classifier.	1.1.0	Runtime 2020.14
/trace/configurations/{classifier}/ /identifiers/	DELETE	Remove all trace identifiers of the the given classifier.	1.1.0	Runtime 2020.14
/trace/configurations/{classifier}/ /identifiers/{id}	DELETE	Remove the trace identifier with the given ID within the the given classifier.	1.1.0	Runtime 2020.14
/trace/configurations/{classifier}/ /identifiers/{id}/raw	GET	Get the tracing data of the identifier with the given ID within the the given classifier as stored internally (raw data).	1.1.0	Runtime 2020.14
/trace/identifiers/	GET	List all asynchronous trace identifiers of the service.	1.1.0	Runtime 2020.14
/trace/identifiers/	DELETE	Remove all asynchronous trace identifiers of the service.	1.1.0	Runtime 2020.14