

## PAPI-WS Extension to retrieve Fuego Instances using Filters

A new method and supporting structures have been added to the PAPI-WS (PAPI Web Services API) to support querying the Fuego Engine for instances matching a particular criteria.

### *New Method*

The new introduces method in PAPI-WS is called “getInstancesUsingFilter”. This method receives the following parameters:

- **sessionId:** This is the session identified randomly generated when creating a session.
- **processId:** This is the process identifier. The search criteria will be run for all instances in the specified Fuego Business Process. The process Id does not need the process version although it can be provided. For process version independency it is recommended, that only the processId is provided (ie: “/ProcessA”).
- **filterAttributes:** This is a list of FilterAttribute objects that will narrow down the collection of instances to be retrieved by the method based on searching conditions.

This method returns a collection of InstanceInfo objects that match the search criteria specified in the incoming filterAttributes.

### *New Supporting Structures*

The incorporation of this new method into PAPI-WS also includes the definition of new structures needed by the method to define the filtering criteria.

Two new structures have been added: FilterAttribute and FilterOperation.

### **FilterAttribute**

This structure has the form of a tuple of 3 attributes:

- **variable:** This is the variable that will define the search criteria.
- **operator:** This is the operator to apply to the specified variable and value attributes. See FilterOperator below.
- **value:** This is the value that will narrow down the search for the specified variable.

Each one of this FilterAttribute objects will define a search criteria filtering when sent to the Fuego Engine to retrieve instances.

The variable names will define the dimension by which the instances in the Fuego Engine should be filtered. The variable name may be one of the existing Fuego Predefined variables such as activity name or one of the process specific created variables as far as they have been define of native types (String, integer, Boolean).

## Process Predefined variable names

This is the list of valid Predefined variable names:

- **PREDEFINE\_ACTIVITY:** This variable specifies what activity within the Fuego Business Process Context the instance is located on.
- **PREDEFINE\_DESCRIPTION:** This variable specifies the description associated to the instance.
- **PREDEFINE\_PARTICIPANT:** This variable specifies the participant that may have exclusive access to the instance within an activity of the Fuego Business Process.
- **PREDEFINE\_STATUS:** This variable specifies the status of the instance within a Fuego Business Process (ie: “In Progress”, “Completed”, “Aborted”).

## Process specific variable names

Process specific variables names are the ones created when deploying a Fuego project.

To use process specific variable names in filters, you need to declare these variables as “External” or as of “Business” type in your Fuego Business Process.

## FilterOperation

This structure basically wraps the operator to use when using a FilterAttribute object. The available operator list is as follows:

- **“IS”:** This operator will look for values matching exactly the value in the “value” field.
- **“IS NOT”:** This operator will look for values not matching the value in the “value” field.
- **“GREATER THAN”:** This operator will look for values that are greater than the value in the “value” field.
- **“GREATER OR EQUAL”:** This operator will look for values that are greater than or equal the value in the “value” field.
- **“LESS THAN”:** This operator will look for values that are less than the value in the “value” field.
- **“LESS OR EQUAL”:** This operator will look for values that are less than or equal than the value in the “value” field.

- **"CONTAINS"**: This operator will look for values that contain the value in the "value" field.
- **"BEGINS WITH"**: This operator will look for values that begin with the value in the "value" field.
- **"ENDS WITH"**: This operator will look for values that end with the value in the "value" field.

## Filtering search semantics

When providing more than one FilterAttribute when invoking the `getInstancesUsingFilter` method the following semantics will apply to run the search criteria

1. If the variable names are all different, then they will be linked using the "AND" operator. For example, if we create a filter with variables "PREDEFINE\_ACTIVITY" and "custId", then the search will be running with the following operators: ("PREDEFINE\_ACTIVITY" =X AND "custId" = Y).
2. For repeated variable names, the "OR" operator will be used. For example, if we create a filter with variables "PREDEFINE\_ACTIVITY" and 2 instances of "custId", then the search will be running with the following operators: ("PREDEFINE\_ACTIVITY = X" AND ("custId" = Y OR "custId" = Z)).

## Example of usage of `getInstanceUsingFilter` method using one FilterAttribute Search Criteria

```
String sessionId = processServiceControl.createSessionWithPreset("tmedic1",
"password", true, "container-auth");

PAPIWS.ProcessServiceControl.FilterAttribute[] filterAttrs = new
PAPIWS.ProcessServiceControl.FilterAttribute[1];
filterAttrs[0] = new PAPIWS.ProcessServiceControl.FilterAttribute();
filterAttrs[0].variable = "PREDEFINE_ACTIVITY";
filterAttrs[0].operator = new PAPIWS.ProcessServiceControl.FilterOperator();
filterAttrs[0].operator.operator = "IS";
filterAttrs[0].value = "FirstStep";
PAPIWS.ProcessServiceControl.InstanceInfo[] res =
processServiceControl.getInstancesUsingFilter(sessionId,
"/ProcessA", filterAttrs);

processServiceControl.close(sessionId);
```

In this case, we are connecting with a user “tmedic1” and password “password” with the “reconnect” attribute set to true. The creation of the session may be done using any of the other session creation methods.

A FilterAttribute is created to search for instances in the activity named “FirstStep” (note that we are using the “IS” operator). When the getInstancesUsingFilter method is invoked, we are specifying that the Filter should be applied to the process named “ProcessA”.

### **Example of usage of getInstanceUsingFilter method using two FilterAttribute Search Criteria**

```
String sessionId = processServiceControl.createSessionWithPreset("eduardoc",
"password", true, "container-auth");

PAPIWS.ProcessServiceControl.FilterAttribute[] filterAttrs = new
PAPIWS.ProcessServiceControl.FilterAttribute[1];
filterAttrs[0] = new PAPIWS.ProcessServiceControl.FilterAttribute();
filterAttrs[0].variable = "PREDEFINE_ACTIVITY";
filterAttrs[0].operator = new PAPIWS.ProcessServiceControl.FilterOperator();
filterAttrs[0].operator.operator = "IS";
filterAttrs[0].value = "FirstStep";
filterAttrs[1] = new PAPIWS.ProcessServiceControl.FilterAttribute();
filterAttrs[1].variable = "custId";
filterAttrs[1].operator = new PAPIWS.ProcessServiceControl.FilterOperator();
filterAttrs[1].operator.operator = "IS";
filterAttrs[1].value = "1";
PAPIWS.ProcessServiceControl.InstanceInfo[] res =
processServiceControl.getInstancesUsingFilter(sessionId,
"/ProcessA", filterAttrs);

processServiceControl.close(sessionId);
```

In this case, we are connecting with a user “tmedic1” and password “password” with the “reconnect” attribute set to true. The creation of the session may be done using any of the other session creation methods.

A FilterAttribute is created to search for instances in the activity named “FirstStep” (note that we are using the “IS” operator). A second FilterAttribute is created to search for the process instance variable custId with a value “1” assigned to it.

When the getInstancesUsingFilter method is invoked, we are specifying that the Filter should be applied to the process named “ProcessA”. As we are specifying 2 FilterAttributes using different variables, the collection returned will be composed of instances that are in the activity “FirstStep” that also have the process instance variable set to “1”.