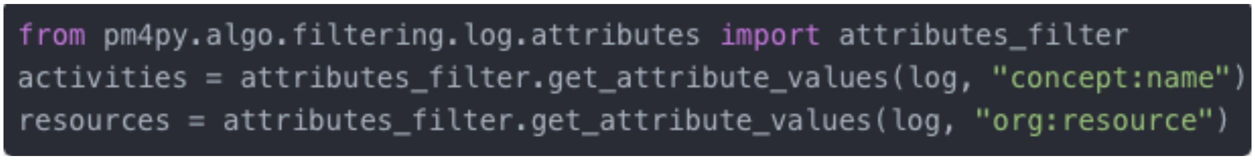
Filtering based on attribute values.

Filtering on attributes values permits alternatively to:

- Keep cases that contains at least an event with one of the given attribute values
- Remove cases that contains an event with one of the the given attribute values
- Keep events (trimming traces) that have one of the given attribute values
- Remove events (trimming traces) that have one of the given attribute values

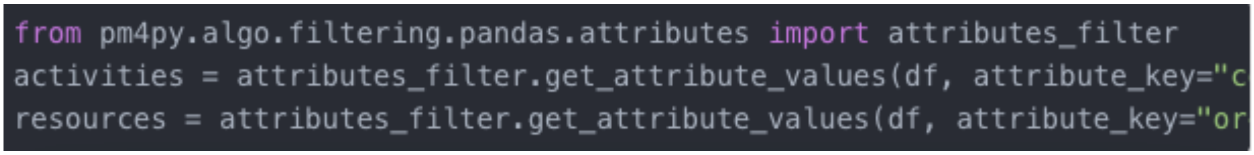
Example of attributes are the resource (generally contained in org:resource attribute) and the activity (generally contained in concept:name attribute). As noted before, the first method can be applied on log objects, the second on dataframe objects.



```
from pm4py.algo.filtering.log.attributes import attributes_filter
activities = attributes_filter.get_attribute_values(log, "concept:name")
resources = attributes_filter.get_attribute_values(log, "org:resource")
```

Figure – 1

To get the list of resources and activities contained in the log, the following code could be used.

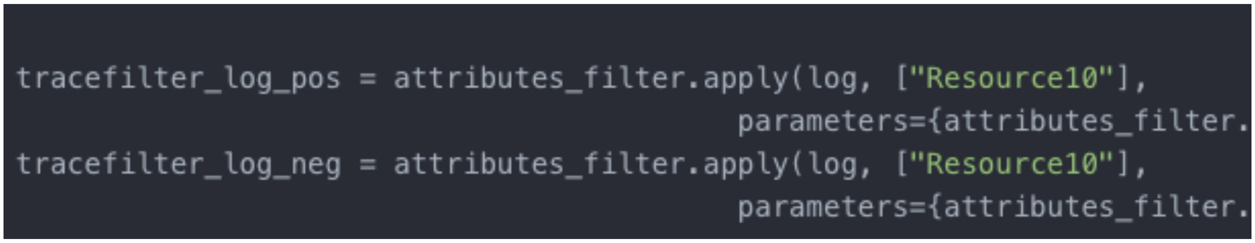


```
from pm4py.algo.filtering.pandas.attributes import attributes_filter
activities = attributes_filter.get_attribute_values(df, attribute_key="c
resources = attributes_filter.get_attribute_values(df, attribute_key="or
```

Figure -2

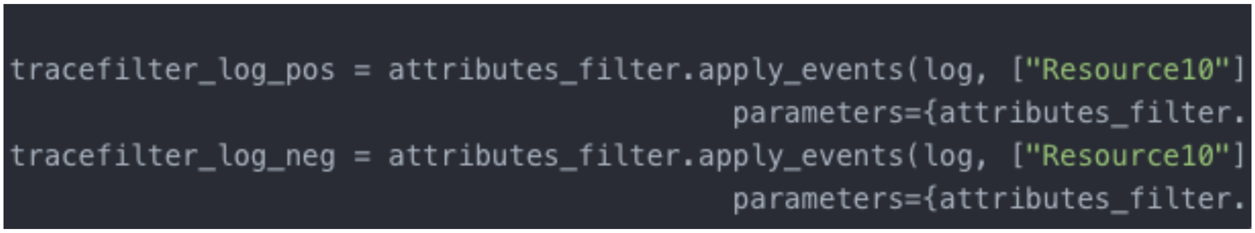
To filter traces containing/not containing a given list of resources, the following code could be used

It is also possible to keep only the events performed by a given list of resources (trimming the cases). The following code can be used from figure 3 and figure 4



```
tracefilter_log_pos = attributes_filter.apply(log, ["Resource10"],
                                          parameters={attributes_filter.
tracefilter_log_neg = attributes_filter.apply(log, ["Resource10"],
                                          parameters={attributes_filter.
```

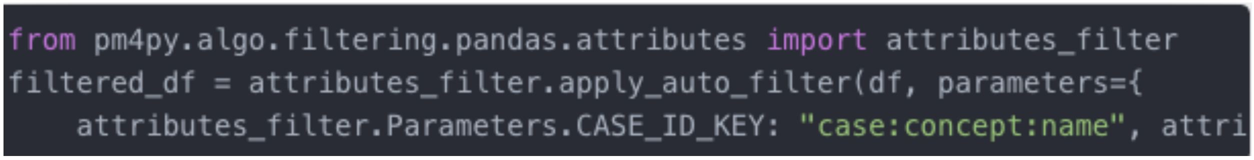
Figure 3



```
tracefilter_log_pos = attributes_filter.apply_events(log, ["Resource10"]
                                          parameters={attributes_filter.
tracefilter_log_neg = attributes_filter.apply_events(log, ["Resource10"]
                                          parameters={attributes_filter.
```

Figure 4

To apply automatically a filter on events attributes (trimming traces and keeping only events containing the attribute with a frequent value), the apply_auto_filter method is provided. The method accepts as parameters the attribute name and the DECREASING_FACTOR (default 0.6; an explanation could be found on the start activities filter).



```
from pm4py.algo.filtering.pandas.attributes import attributes_filter
filtered_df = attributes_filter.apply_auto_filter(df, parameters={
    attributes_filter.Parameters.CASE_ID_KEY: "case:concept:name", attri
```

Figure 5