

Adding Ad Hoc Filters to the Java Engine

Overview

New in version 11.0.

Our engine API supports filters in the select. It moves the filtering logic from the template to the API creating a specific select for each query.

What this Functionality Does

This allows you to can change what data is displayed in the generated report. You can create one template and then, based on the value of variables passed to the datasource when running the report, set what rows of data are rendered in the report. For example, you may have a table listing all sales for your company. You can then pass in filters to limit the table to sales in the western region, over the previous quarter. The next run may be all sales for one sale rep. You can filter any way you wish without ever changing the template.

For example, you may have a table listing all sales for your company. You can then pass in filters to limit the table to sales in the western region, over the previous quarter. The next run may be all sales for one sale rep. You can filter any way you wish without ever changing the template.

The Code

See the file [FilterVariables_Java.zip](#) attached to this article. This is also installed as a sample program in the java engine samples directory.

Here's the additional code to select several specific rows of data:

```
Map<String, Object> adHocVariables = new HashMap<String, Object>();
List<Serializable> list = new ArrayList<Serializable>();
list.add(1);
list.add(2);
adHocVariables.put("employeeId", new SelectList(SelectBase.SORT_NO_OVERRIDE,
"employeeId", list));
datasource.setMap(adHocVariables);
report.processData(datasource, "");
```

And here is the code for a condition:

```
Map<String, Object> adHocVariables = new HashMap<String, Object>();
```

```
adHocVariables.put("employeeId", new SelectFilter(SelectBase.SORT_NO_OVERRIDE,
"employeeId",
new SelectFilter.Filter[]{new SelectFilter.Filter("@EmployeeID",
SelectFilter.Filter.OP_EQUALS, 1)},
true));
datasource.setMap(adHocVariables);
report.processData(datasource, "");
```

It makes it incredibly easy to use this new functionality. And I think this will become very popular (essential for many). Because it moves the filtering logic from the template, and removes it from creating a specific select for each query, and instead leaves it that you just define the filter and we take care of the rest.