**DQLUtils xCelerator**

The DQLUtils xCelerator provides the ability to run one or several read, execute and apply DQL queries in one activity with fully dynamic queries, including support for IN clauses, built up from a template, process parameters and process variables or entirely defined in either process parameters or process variables.

Often as part of the setup of a process it is necessary to retrieve different configuration data, it can often result in a sequence of DQL Read activities.

The xCelerator dqlutils project contains a single stateless process with a single Java activity configured to use the com.emc.xcelerators.community.dqlutils.DQLUtilsImpl class. The default sample page for the project includes an example of the use of the process.

**Overview of the Methods DQLUtils Provides**

The DQLUtils methods all perform one or several queries on the current repository. The read, execute and apply are all convenience methods that invoke the more generic query method. All methods also exists in two version, one which take one query and return one result set and one which takes multiple queries (an array of QueryTemplate) and return multiple result sets (an array of ResultSet).

**Methods**

ResultSet **apply**(QueryTemplate query) ResultSet[] **apply**(QueryTemplate[] query) ResultSet **exec**(QueryTemplate dqlQuery) ResultSet[] **exec**(QueryTemplate[] dqlQuery) ResultSet **read**(QueryTemplate query) ResultSet[] **read**(QueryTemplate[] query)

ResultSet **query**(QueryTemplateWithOptions query)

ResultSet[] **query**(QueryTemplateWithOptions[] query)

**QueryTemplate**

| Parameter Type Description | | |
| --- | --- | --- |
| template | String | The query or query template. Any occurrence of  $<number>$, i.e. $0$, $1$ etc are replaced by the value of the corresponding TemplateParam. |
| params | TemplateParam | The template param denotes a value or a number of values that are to be placed into the template to yield the final query. There are three options that are available:  **escape**, if set to true will escape any single quotes by doubling them (i.e. “It’s” will become “It’’s”).  **quote**, if set to true will surround the value(s) with single quotes.  **concat**, if set to true will concatenate all values into one  value (optionally separated by a separator), if set to false will separate all values with a comma.  separator, if concat is set to true and multiple values are provided, concat will use the string as a separator in between each value, default is the empty string. |

**QueryTemplateWithOptions**

Extends QueryTemplate with the addition of an option parameter, for details regarding template and params see QueryTemplate.

|  |  |  |
| --- | --- | --- |
| Parameter Type Description | | |
| option | String | A string denoting what type of query, valid values are: APPLY for an apply query.  READ for a read query  EXEC for a exec query |

**ResultSet**

The ResultSet type is used for passing the result of a query, regardless of type, back to the caller so it can be used in the output mapping of the BOF Module activity. The result set consists of an array of rows and a row consists of an array of columns and each column is a string value.

**Considerations**

**Logging**

If you have trouble getting your queries correct, you can turn on DEBUG level logging for the category com.emc.xcelerators.community.dqlutils.DQLUtilsImpl in your log4j.properties (or equivalent) and the complete queries will be output to the log. Do note that you should only use DEBUG level logging on a development system since significant amount of data will be added to the log.