

Load Grid Data with Custom Sorting and Filtering

In this article, we are going to use Bean Shell code to populate our grid with a custom order and filter out some unwanted data. There are 2 ways to achieve this.

1. Using FormDataDao to retrieve data

`FormDataDao` is a service class dealing with form data. We will be using the find method to retrieve data and using the `sort` parameter to sort the data. We can use the `condition` & `params` parameters to filter it as well.

```
public org.joget.apps.form.model.FormRowSet find(java.lang.String formDefId, java.lang.String tableName,
final java.lang.String condition, final java.lang.Object[] params, final java.lang.String sort, final java.
lang.Boolean desc, final java.lang.Integer start, final java.lang.Integer rows)
```

Sample Code:

```
import org.joget.apps.app.service.AppUtil;
import org.joget.apps.form.dao.FormDataDao;
import org.joget.apps.form.model.Element;
import org.joget.apps.form.model.FormData;
import org.joget.apps.form.model.FormRow;
import org.joget.apps.form.model.FormRowSet;
import org.joget.apps.form.service.FormUtil;
import org.joget.plugin.base.PluginManager;
import org.joget.apps.form.model.FormLoadBinder;
import org.joget.commons.util.LogUtil;

public FormRowSet load(Element element, String primaryKey, FormData formData) {
    String formDefId = "grid_entry"; //change this to the form id used to store grid data
    String tableName = "grid_entry"; //change this to the table name used to store grid data
    String foreignKey = "fk"; //change this to the foreign key
    String sort = "name"; //change to other field id for sorting
    Boolean desc = false; //change to true for descending order

    FormRowSet rows = new FormRowSet();

    try {
        FormDataDao formDataDao = (FormDataDao) AppUtil.getApplicationContext().getBean("formDataDao");
        String foreignKeyFilter = getFormPropertyName(foreignKey);
        String condition = (foreignKeyFilter != null && !foreignKeyFilter.isEmpty()) ? " WHERE " +
foreignKeyFilter + " = ?" : "";
        Object[] paramsArray = new Object[]{primaryKey};

        //add additional filter
        /*
            String nameFilter = getFormPropertyName("name");
            condition += " AND " + nameFilter + " LIKE ?";
            paramsArray = new Object[]{primaryKey, "%TEST%"};
        */

        rows = formDataDao.find(formDefId, tableName, condition, paramsArray, sort, desc, null, null);
    } catch (Exception e) {
        LogUtil.error("Bean Shell Form Load Binder", e, "Load data with custom sorting and filtering");
    }
    rows.setMultiRow(true);
    return rows;
}

public String getFormPropertyName(String propertyName) {
    if (propertyName != null && !propertyName.isEmpty()) {
        if (!FormUtil.PROPERTY_ID.equals(propertyName)) {
            propertyName = FormUtil.PROPERTY_CUSTOM_PROPERTIES + "." + propertyName;
        }
    }
    return propertyName;
}

//call load method with injected variable
return load(element, primaryKey, formData);
```

2. Post Processing the Data

In this method, we will reuse the multirow form binder to retrieve the data then do some post processing to sort or filter it before return the data to the grid.

```
import org.joget.apps.form.model.Element;
import org.joget.apps.form.model.FormData;
import org.joget.apps.form.model.FormRow;
import org.joget.apps.form.model.FormRowSet;
import org.joget.apps.form.service.FormUtil;
import org.joget.plugin.base.PluginManager;
import org.joget.apps.form.model.FormLoadBinder;
import java.util.Collections;
import java.util.Comparator;

public FormRowSet load(Element element, String primaryKey, FormData formData) {
    String formDefId = "grid_entry"; //change this to the form id used to store grid data
    String foreignKey = "fk"; //change this to the foreign key
    final String sortField = "name";

    FormRowSet f = new FormRowSet();

    // Reuse Multi Row Binder to load data
    PluginManager pluginManager = (PluginManager) FormUtil.getApplicationContext().getBean("pluginManager");
    FormLoadBinder binder = (FormLoadBinder) pluginManager.getPlugin("org.joget.plugin.enterprise.
MultirowFormBinder");

    //Load from the grid table
    binder.setProperty("formDefId", formDefId);
    binder.setProperty("foreignKey", foreignKey);
    f = binder.load(element, primaryKey, formData);

    //post processing
    if (f != null && !f.isEmpty()) {
        Collections.sort(f, new Comparator() {
            public int compare(Object row1, Object row2) {
                String sort1 = ((FormRow) row1).getProperty(sortField);
                String sort2 = ((FormRow) row2).getProperty(sortField);
                return sort1.compareTo(sort2);
            }
        });
    }
    return f;
}

//call load method with injected variable
return load(element, primaryKey, formData);
```

The sample app for this article: [APP_custom_grid_load-1-20151119111806.jwa](#)