Table of Contents

With JVx, it's very easy to define dropdown list boxes, LOVs, or picklists. JVx supports manual definition or automatic definition of dropdowns. If you want to use automatic dropdowns, read the following:

Automatic List Boxes based on the DB Model.

If you want to define custom dropdowns, continue reading.

JVx offers different options with or without database.

UIEnumCellEditor

The easiest solution is an UIEnumCellEditor.

```
UIEnumCellEditor cellEditor = new UIEnumCellEditor();
cellEditor.setAllowedValues(new String[] {"A", "B", "C", "D"});
cellEditor.setDisplayValues(new String[] {"Alpha", "Bravo", "Charlie",
"Delta"});
```

The cell editor saves an allowed value and displays a display value.

After the definition, set the cell editor to the datatype of a specific column.

```
dataBook_or_row.getRowDefinition().getColumnDefinition("VALUE").
                                    getDataType().setCellEditor(cellEditor);
```

Use an IDataBook or IDataRow. An IDataBook is, for example, the MemDataBook or the RemoteDataBook. An IDataRow is, for example, the DataRow.

It's possible to create a DataRow from an existing IDataBook, e.g.:

```
IDataRow row = baseDataBook.createEmptyRow(new String[] {"LINK_ID",
"LINK_VALUE"});
```

or to create a new DataRow.

```
RowDefinition rowdef = new RowDefinition();
rowdef.addColumnDefinition(new ColumnDefinition("LINK_ID", new
BigDecimalDataType()));
rowdef.addColumnDefinition(new ColumnDefinition("LINK_VALUE"));

DataRow row = new DataRow(rowdef);
```

To use automatic translation, follow these instructions.

UILinkedCellEditor

If you need more flexibility, use an UILinkedCellEditor for your dropdowns. The UIEnumCellEditor maps a value to one ID column and displays the value column.

The UILinkedCellEditor shows one or more columns and maps multiple ID columns. The cell editor needs an IDataBook which contains the data. Use a MemDataBook or a RemoteDataBook. Our example uses a MemDataBook.

```
RowDefinition rowdef = new RowDefinition();
rowdef.addColumnDefinition(new ColumnDefinition("ID", new
BigDecimalDataType()));
rowdef.addColumnDefinition(new ColumnDefinition("VALUE"));

MemDataBook memBook = new MemDataBook(rowdef);
memBook.setName("memory");
memBook.open();

UILinkedCellEditor linkCellEditor = new UILinkedCellEditor();
linkCellEditor.setLinkReference(new ReferenceDefinition(
                      new String[] {"LINK_ID", "LINK_VALUE"},
                      memBook,
                      new String[] {"ID", "VALUE"}));

dataBook_or_row.getRowDefinition().getColumnDefinition("LINK_VALUE").
                                   getDataType().setCellEditor(cellEditor);
```

We map the ID to LINK_ID and VALUE to LINK_VALUE. The VALUE column will be shown in the dropdown. After selecting a value, the ID and VALUE will be written to LINK_ID and LINK_VALUE.

You can use the UILinkedCellEditor with a RemoteDataBook and read record from a remote storage, e.g., a database, a XML file, CSV file, or any other datasource. You don't need a specific datasource if you use a MemDataBook.

From:
http://doc.sibvisions.com/ - **Documentation**

Permanent link:
http://doc.sibvisions.com/jvx/client/gui/custom_linked_celleditor

Last update: **2020/06/26 11:54**