**Using JXTreeTable to display grouped data**

Let me start by ranting about Swing. Its a lame duck. It could never really take off. But we can't live with it, can't live without it. Especially if the application is a legacy Swing application written 10 years ago.

So, when I had the prospect of displaying simple grouped data, the fear of writing my custom JTree absolutely seized me. That is when i thought about the SwingX project. In its present form, it buggy, unusable, messed up etc. It took me close to an hour to figure out a stable release from the quagmire of broken links that Google came up with. At last, the Maven Central came to my rescue in the form of the following 3 magical lines:

<dependency>

<groupId>org.swinglabs.swingx</groupId>

<artifactId>swingx-all</artifactId>

<version>1.6.4</version>

</dependency>

**The First 90%**

After being associated with Swing for a good part of my life, I have come to believe that 90% of your job is done if you can create a good Model, especially for unwieldy components like JTable and JTree. So, I do the same for this as well. In this case, I represent grouped data with a Map<String, List<BankDetail>>. All I want is to display the groups as nodes and the Lists under these nodes. This is how I model it:

**public** **class** BankDetailTreeTableModel **extends** AbstractTreeTableModel {

**private** **static** **final** String[] ***COLUMN\_NAMES*** = **new** String[] { "id", "age",

"job", "marital", "education", "default", "balance" };

**private** **static** **final** String ***ROOT*** = "\_ROOT\_";

**private** **final** Map<String, List<BankDetail>> groupedBankDetails;

**private** **final** List<String> groups;

**public** BankDetailTreeTableModel(

Map<String, List<BankDetail>> groupedBankDetails) {

**super**(***ROOT***);

**this**.groupedBankDetails = groupedBankDetails;

groups = getGroups(groupedBankDetails);

}

@Override

**public** **int** getColumnCount() {

**return** ***COLUMN\_NAMES***.length + 1;

}

@Override

**public** String getColumnName(**int** column) {

**if** (column == 0) {

**return** "Group";

}

**return** ***COLUMN\_NAMES***[column - 1];

}

@Override

**public** Object getValueAt(Object node, **int** column) {

**if** (node **instanceof** String) {

**if** (column == 0) {

**return** node;

}

**return** ***COLUMN\_NAMES***[column - 1];

} **else** **if** (node **instanceof** BankDetail) {

**if** (column == 0) {

**return** **null**;

}

**return** displayColumnValue((BankDetail) node, column - 1);

}

**return** **null**;

}

@Override

**public** Object getChild(Object parent, **int** index) {

**if** (***ROOT***.equals(parent)) {

**return** groups.get(index);

} **else** **if** (parent **instanceof** String) {

**return** groupedBankDetails.get(parent).get(index);

}

**return** **null**;

}

@Override

**public** **int** getChildCount(Object parent) {

**if** (***ROOT***.equals(parent)) {

**return** groups.size();

} **else** **if** (parent **instanceof** String) {

**return** groupedBankDetails.get(parent).size();

}

**return** 0;

}

@Override

**public** **int** getIndexOfChild(Object parent, Object child) {

**return** 0;

}

**private** List<String> getGroups(

Map<String, List<BankDetail>> groupedBankDetails) {

List<String> groups = **new** ArrayList<>(groupedBankDetails.keySet());

Collections.*sort*(groups);

**return** groups;

}

**private** String displayColumnValue(BankDetail bankDetail, **int** columnIndex) {

**switch** (columnIndex) {

**case** 0:

**return** Integer.*toString*(bankDetail.getId());

**case** 1:

**return** Integer.*toString*(bankDetail.getAge());

**case** 2:

**return** bankDetail.getJob();

**case** 3:

**return** bankDetail.getMarital();

**case** 4:

**return** bankDetail.getEducation();

**case** 5:

**return** bankDetail.getDefaulted();

**case** 6:

**return** bankDetail.getBalance().toPlainString();

**default**:

**throw** **new** IllegalArgumentException("columnIndex " + columnIndex

+ " is not handled");

}

}

}

**The remaining 10%**

The tying up the model to the JXTreeTable is the simplest thing on the earth:

Map<String, List<BankDetail>> groupedBankDetails = bankDetailServiceToUse

.getBankDetails(selectedGroup);

treeTblBankDetails.setTreeTableModel(**new** BankDetailTreeTableModel(

groupedBankDetails));

The source code can be found here:

<https://github.com/paawak/blog/tree/master/code/lambda-demo>

The main class to run is: GroupingDemoFrame.

I found the following post very useful in my efforts:

<http://javanbswing.blogspot.com/2013/08/swing-treetable-example-using.html>