**Table Multi Select**

**Step 1:**

* Drag and drop the table component onto your page designer.
* Use the Quick Start Method to assign table data.
* Customize the display and style of the table.
* In the table properties, select 'Multiple' in the 'Selection Mode' for rows.
* Add an ID for the table.

A screen shot of a computer

Description automatically generated

**Step 2:**

Create these variables in a page:

* Create a variable named **currentSelection** with a data type of String.
* Create a variable named **disableDeselectAll** with a data type of Boolean and a default value of true.
* Create a variable named **rowData** with a data type of Array.

**Step 3:**

* The JavaScript code defines a module using Knockout and Oracle JET KeySet.
* It sets up the functionality to manage selected rows in the table.
* The selectedChangedListener function handles changes in row selections and provides a message describing the selection.
* Other utility functions like deselectAll, selectSpecificRows, and getRowsForIDs are used for managing selections and retrieving data.

A screenshot of a computer

Description automatically generated

Copy & paste below Javascript Code.



**STEP 4:**

Select the table and create the '**On Select Multiple Rows'** event.

A screenshot of a computer

Description automatically generated

**Step 4.1:**

The rowData variable is utilized to store the values of the selected rows.

So first, **Reset** the **rowData** Variable

A screenshot of a computer

Description automatically generated

**Step 4.2:**

Drag & Drop the Assign Variable.

* This step assigns the function **selectedChangedListener** to the **currentSelection** variable.
* The function generates a message based on the selected rows in the table.

A screenshot of a computer

Description automatically generated

* If you select all rows in the table, the code says, "You've selected all the rows."
* If you initially select all rows by checking a "Select All" option, but then unselect some, it says, "You've picked all rows except for the ones you removed."
* If you only pick specific rows within the table, it tells you exactly which rows you've chosen.
* Ultimately, it provides you with a message that explains your current selection.

**Step 4.3:**

Drag & Drop the Assign Variable.

* This step assigns the **isSelectionEmpty** function to the **disableDeselectAll** variable.
* This function checks if any rows are currently selected in a table. If it finds that at least one row is selected, it returns "false," indicating that there are selected rows. If it doesn't find any selected rows, it returns "true," indicating that no rows are currently selected.

A screenshot of a computer

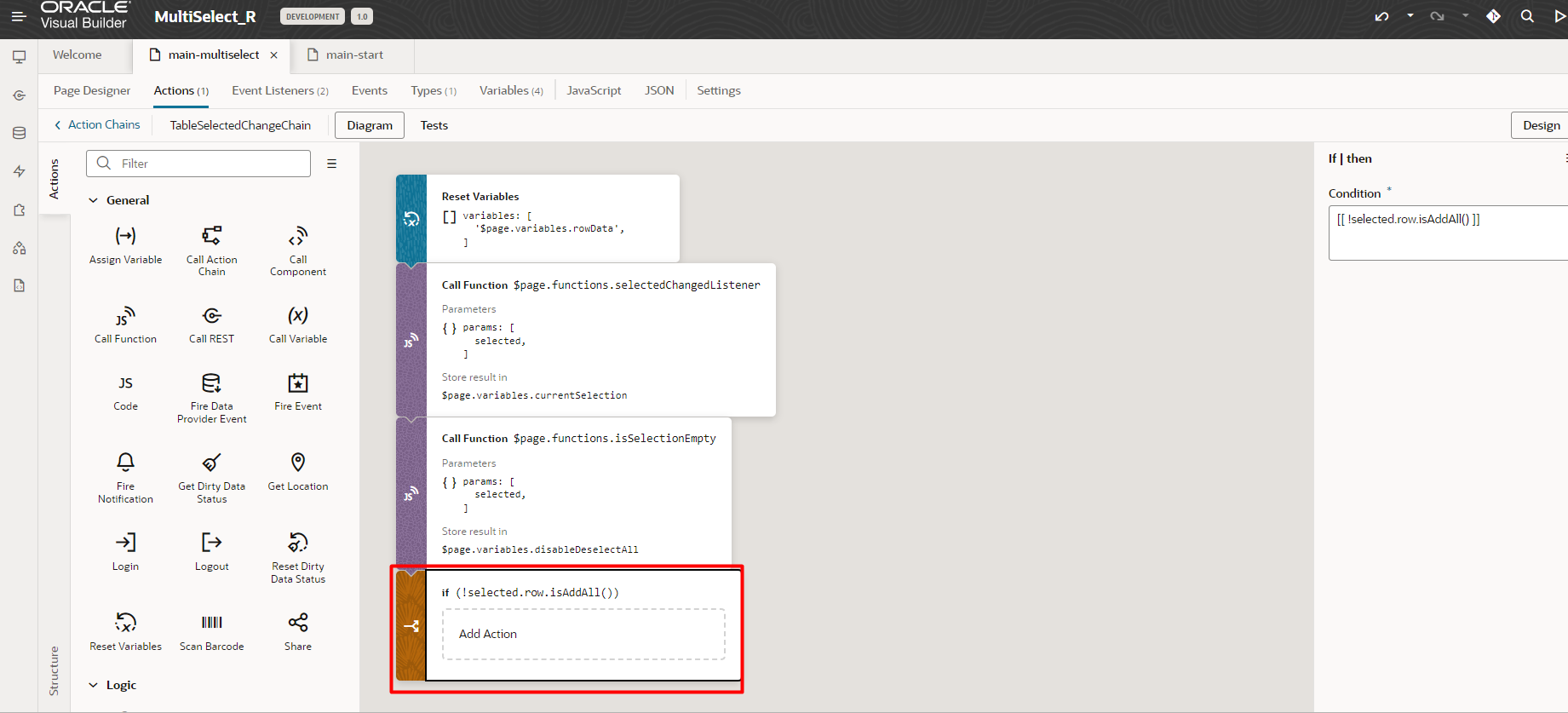
Description automatically generated

**Step 4.4:**

Add If Condition

**selected.row.isAddAll()**

* When the header checkbox is selected, the function returns true.
* The header checkbox is selected, and the data row checkboxes are deselected, resulting in the function returning true.
* Otherwise the function returns false.



**STEP 4.5:**

If selected.row.isAddAll() returns false, Assign the result of the **getRowsForIDs** JavaScript function to the **rowData** variable

A screenshot of a computer

Description automatically generated

table - document.getElementById('multiSelectTableID')

rowIDs - Array.from(selected.row.values())

A screenshot of a computer

Description automatically generated

Imagine you're in a room full of toys, and you have a checklist of specific ones you want. This code is your guide:

You start at the first toy.

You have a checklist of toy names.

You check each toy's name.

If it matches, you pick it up, mark it off, and remove it.

You continue until two things happen:

You've checked 1000 toys.

You've found all on your list (your list is empty).

After, you show what you collected.

In the code:

table is the room with toys.

rowIDs is your checklist.

index is where you are.

result is your bag for collected toys.

The loop checks each toy.

getDataForVisibleRow(index) checks the toy's name.

The if statements decide what to collect.

The return statements stop you, whether you find all or reach 1000.

After, you show what's in your bag (result).

**STEP 5:**

In the Properties pane of the Table component, assign [[$page.functions.getSelectedRows()]] to the selected.row attribute.

A screenshot of a computer

Description automatically generated

**STEP 6:**

Drag & Drop One Button & 2 text area component.

Modify their styles, place holder.

A screenshot of a computer

Description automatically generated

**STEP 7:**

Current Selection – Text Area Component

Assign Value

value="{{ ($variables.currentSelection ? $variables.currentSelection :'')  + ($variables.rowData.length === 0 ? '' : '\n'+ 'Row Data: '+JSON.stringify ($variables.rowData)) }}"

* It checks if a variable called $variables.currentSelection has a value. If it does, it uses that value. If it doesn't, it uses an empty string as the default value.
* It then adds more to the value based on the length of an array called $variables.rowData:
* If $variables.rowData is an empty array (it has no items), it adds nothing to the value.
* If $variables.rowData has items, it adds a newline character ('\n'), followed by the text 'Row Data: ', and then the JSON representation of $variables.rowData.

A screenshot of a computer

Description automatically generated

**STEP 8:**

Select rows with property IDs – Text Area Component

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Call Javascript Function – selectSpecificRows

Pass its value as rowIdsString

* The selectSpecificRows function takes a list of row IDs as a string (e.g., "1,2,3"). It splits the string into individual IDs, converts them into numbers, and then selects those rows in a table.

A screenshot of a computer

Description automatically generated

**STEP 9:**

Deselect All – Button

* Disable condition – disableDeselectAll Variable is true, this button will disabled.
* Whenever there is no selections in the table, disableDeselectAll Variable get true value. We assigned it in TableSelectedChangeChain Event.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Call the **deselectAll** Javascript Function. It clears all selected rows in a table. When you call this function, it effectively deselects or unchecks all rows that were previously selected.

