

Liquid UI: Creating a Class

In this example, a Class called **Item** will be created. This class will have attributes like **Material** and **Quantity**. Using a class to hold this data helps the code be more modular and makes it easier to access data pertaining to a single item.

User Interface

Log in to SAP and navigate to the transaction VA03.

The screenshot shows the SAP VA03 'Display Sales Order: Initial Screen' interface. At the top, there is a menu bar with 'Sales document', 'Edit', 'Goto', 'Environment', 'System', and 'Help'. Below the menu bar is a toolbar with various icons for navigation and actions. The main title of the screen is 'Display Sales Order: Initial Screen'. Below the title, there is a navigation bar with icons for 'Sales', 'Item overview', 'Ordering party', 'Orders', and a document icon. The main area of the screen contains a form for entering sales order data. The 'Order' field is highlighted with a red box and contains the value '6666'. To the right of the 'Order' field is a 'Display items' button. Below the 'Order' field is a 'Search Criteria' section with a table of search fields: 'Purchase Order No.', 'Sold-to party', 'Delivery', 'Billing Document', and 'WBS Element'. Each field has a corresponding input box. At the bottom of the 'Search Criteria' section is a 'Search' button with a magnifying glass icon.

Search Criteria	
Purchase Order No.	<input type="text"/>
Sold-to party	<input type="text"/>
Delivery	<input type="text"/>
Billing Document	<input type="text"/>
WBS Element	<input type="text"/>

Enter and order number and then click the pushbutton “Display items”. A function will run, fetching the data from the table of the order you entered. Then the data will be displayed to the user on the initial screen.

Sales document Edit Goto Environment System Help

Display Sales Order: Initial Screen

Sales

Item overview

Ordering party

Orders

Order

Display items

Search Criteria

Purchase Order No.

Sold-to party

Delivery

Billing Document

WBS Element

Search

Line Items

Item	Material	Quantity	SU
10	R100000	21	PC
20	R100001	2	PC
30	R100003	3	PC
40	R100022	3	PC

<

>

...

<

>

Liquid UI Code [Script]

```

1  //////////////////////////////////////
2  // Author: Synactive, Inc. [1065 E. Hillsdale Blvd, Foster City, CA, 94404, USA]
3  // Email: support@guixt.com; sales@guixt.com;
4  // Contact: 650.341.3310
5  //////////////////////////////////////
6
7  // This function creates a class for Item
8  function Item(itm,mat,qty,su){
9      // Attributes of the class
10     this.item = itm;
11     this.material = mat;
12     this.quantity = qty;
13     this.sales_unit = su;
14     // Function used to retrieve information on Item
15     this.getInfo = function(){
16         return "Item:"+this.item+", Material:"+this.material+", Quantity:"+this.quantity+", Sales Unit:"+this.sales_unit;
17     }
18 }
19
20
21
22 // Only execute the following for VA03
23 if(transaction == "VA03"){
24     // Pushbutton that will trigger the function
25     pushbutton("F[Order]+[0,62]", "Display items", "?", {"process":fetchItems});
26
27     // If item_array is undefined, initialize it
28     if(!item_array){
29         item_array = [];
30     }
31
32     // If the array has data in it
33     if(item_array.length>0){
34         // Create a table and columns
35         table([16,0], [26,50], {"name":"z_table", "title":"Line Items", "rows":item_array.length});
36         column("Item", {"size":6,"name":"z_item", "table":"z_table"});
37         column("Material", {"size":18,"name":"z_mat", "table":"z_table"});
38         column("Quantity", {"size":15,"name":"z_qty", "table":"z_table"});
39         column("SU", {"size":3,"name":"z_su", "table":"z_table"});
40
41         // Fill out the table
42         for(i=0;i<item_array.length;i++){
43             println(item_array[i].getInfo());
44             z_table.z_item[i] = item_array[i].item;
45             z_table.z_mat[i] = item_array[i].material;
46             z_table.z_qty[i] = item_array[i].quantity;

```

JavaScript file

length:

```
C:\GuiXT\Tutorials\Forum\SAPMV45A.E0102.sjs - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
SAPMV45A.E0102.sjs
42     for(i=0;i<item_array.length;i++){
43         println(item_array[i].getInfo());
44         z_table.z_item[i] = item_array[i].item;
45         z_table.z_mat[i] = item_array[i].material;
46         z_table.z_qty[i] = item_array[i].quantity;
47         z_table.z_su[i] = item_array[i].sales_unit;
48     }
49
50 }
51
52
53
54
55 // This function will table scroll through the VA03 transaction and fetch the data from the table
56 function fetchItems(){
57     onscreen 'SAPMV45A.0102'
58     enter();
59     onerror
60     message(_message);
61     enter("?");
62     goto FUNC_END;
63     onscreen 'SAPMV45A.4001'
64     // Clear the value of item_array
65     item_array = [];
66     absrow = 1;
67     enter("/ScrollToLine=&V[absrow]", {"table": "T[All items]"});
68
69     NEW_SCREEN:;
70     onscreen 'SAPMV45A.4001'
71     gettableattribute("T[All items]", {"firstvisiblerow": "FVR", "lastvisiblerow": "LVR", "lastrow": "LR"});
72     relrow = 1;
73     NEW_ROW:;
74     println("absrow:"+absrow+", LVR:"+LVR+", LR:"+LR);
75     // end of table?
76     if(absrow>LR){
77         goto END_OF_TABLE;
78     }
79     // end of screen?
80     if(absrow>LVR) {
81         goto NEW_SCREEN;
82     }
83
84     set("V[z_temp_item]", "&cell[All items,Item,&V[relrow]]");
85     set("V[z_temp_mat]", "&cell[All items,Material,&V[relrow]]");
86     set("V[z_temp_qty]", "&cell[All items,Order Quantity,&V[relrow]]");
87     set("V[z_temp_su]", "&cell[All items,SU,&V[relrow]]");

```

JavaScript file

```
C:\GuiXT\Tutorials\Forum\SAPMV45A.E0102.sjs - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
SAPMV45A.E0102.sjs
60      message(_message);
61      enter("?");
62      goto FUNC_END;
63      onscreen 'SAPMV45A.4001'
64      // Clear the value of item_array
65      item_array = [];
66      absrow = 1;
67      enter("/ScrollToLine=&V[absrow]", {"table": "T[All items]"});
68
69  NEW_SCREEN:;
70      onscreen 'SAPMV45A.4001'
71      gettableattribute("T[All items]", {"firstvisiblerow": "FVR", "lastvisiblerow": "LVR", "lastrow": "LR"});
72      relrow = 1;
73  NEW_ROW:;
74      println("absrow:"+absrow+", LVR:"+LVR+", LR:"+LR);
75      // end of table?
76      if(absrow>LR){
77          goto END_OF_TABLE;
78      }
79      // end of screen?
80      if(absrow>LVR) {
81          goto NEW_SCREEN;
82      }
83
84      set("V[z_temp_item]", "&cell[All items,Item,&V[relrow]]");
85      set("V[z_temp_mat]", "&cell[All items,Material,&V[relrow]]");
86      set("V[z_temp_qty]", "&cell[All items,Order Quantity,&V[relrow]]");
87      set("V[z_temp_su]", "&cell[All items,SU,&V[relrow]]");
88
89      // Push a new Item to the array
90      item_array.push(new Item(z_temp_item,z_temp_mat,z_temp_qty,z_temp_su));
91
92      absrow++;
93      relrow++;
94      goto NEW_ROW;
95
96  END_OF_TABLE:;
97
98      enter('/ScrollToLine=1', {"table": "T[All items]"});
99
100     onscreen 'SAPMV45A.4001'
101     enter("/3")
102
103  FUNC_END:;
104 }
```

JavaScript file