



Version 1.0
Date 11.07.2016
Status In Development
Number 4
Initial version by Igor Muntoreanu

Exercise Review:

It will be an Application with excel upload functionality:

		Browse...	Upload					
	First name	User	Last name	City	Street	Country	Region	E-Mail Address
	Igor	IMUNTORE	Muntoreanu	Blumenau	Ingo Hering	BR	BR	igor.muntorea@t-systems.com.br
	Fernando	FBUGMANN	Bugmann	Blumenau	Ingo Hering	BR	BR	fernando.bugmann@t-systems.com.br
	Chuan	CHUANG	Huang	Darmstadt	Pfnorstraße	DE	DE	Chuan.Huang@t-systems.com

Practical Guide:

- 1) Go to our ALV WD Component and navigate to the NODE in the COMPONENTCONTROLLER:

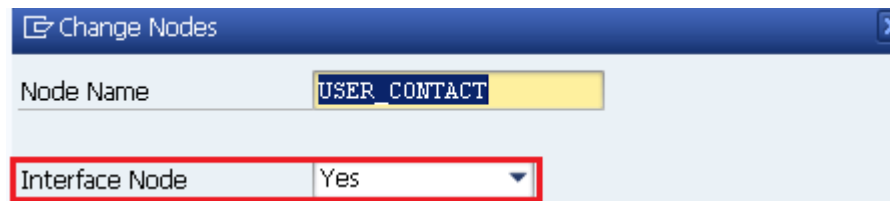
Change the node by right clicking on this:

The screenshot displays the SAP Web Dynpro IDE interface. On the left, the 'Connectivity Browser' shows the project structure for 'ZWDC_USER_DISPLAY_IMUNTOR'. The 'Component Controller' is selected, and the 'Context' tab is active. The context tree shows 'CONTEXT' with a child node 'USER_CONTACT'. A 'Change Nodes' dialog is open, allowing the user to modify the 'USER_CONTACT' node. The dialog fields are as follows:

- Node Name: USER_CONTACT
- Interface Node: No
- Input Element (Ext.): No
- Dictionary structure: (empty field)
- Cardinality: 1..1
- Selection: 0..1
- Init. Lead Selection: Yes
- Singleton: Yes
- Supply Function: (empty field)
- Mapping Path: (empty field)

The 'Property' section at the bottom shows the 'Nodes' property. The 'Add Attributes from Structure' checkbox is checked.

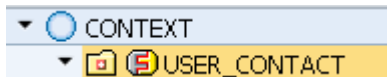
Set the Interface NODE as “Yes”:



The 'Change Nodes' dialog box has a title bar with a blue gradient and a close button. It contains two input fields. The first field, labeled 'Node Name', has the text 'USER_CONTACT' and a yellow background. The second field, labeled 'Interface Node', has a dropdown menu showing 'Yes' and is highlighted with a red border.

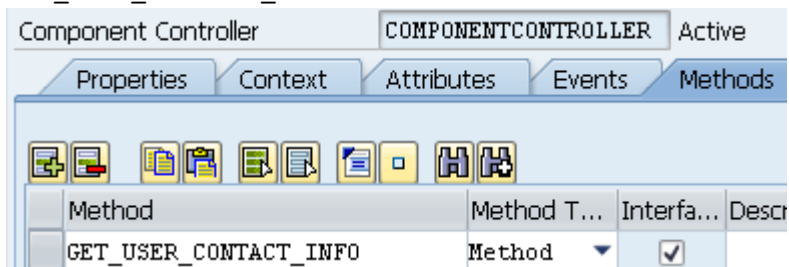
Save and activate

Your node will look like this



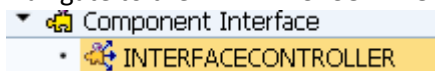
A snippet of a node tree showing a parent node 'CONTEXT' with a blue circle icon. It has a child node 'USER_CONTACT' with a yellow folder icon and a red 'S' icon.

Go to the methods in and mark the “Interface” column of the method
GET_USER_CONTACT_INFO



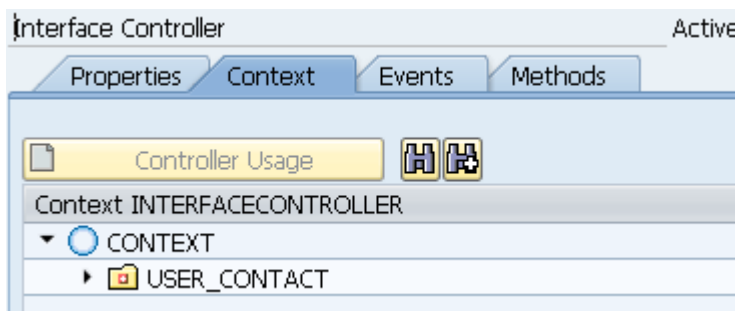
The 'Component Controller' window shows the 'METHODS' tab. The title bar includes 'COMPONENTCONTROLLER' and 'Active'. The tab bar has 'Properties', 'Context', 'Attributes', 'Events', and 'Methods'. Below the tab bar is a toolbar with icons for adding, deleting, and editing methods. A table lists methods with columns 'Method', 'Method T...', 'Interfa...', and 'Descr'. The method 'GET_USER_CONTACT_INFO' is selected, and its 'Interfa...' column contains a checked checkbox.

Navigate to the INTERFACECONTROLLER:

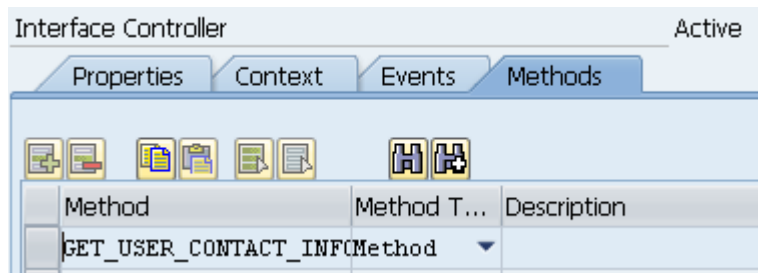


A snippet of a node tree showing a parent node 'Component Interface' with a yellow folder icon. It has a child node 'INTERFACECONTROLLER' with a blue folder icon and a red 'S' icon.

Now we have the context and the method there:

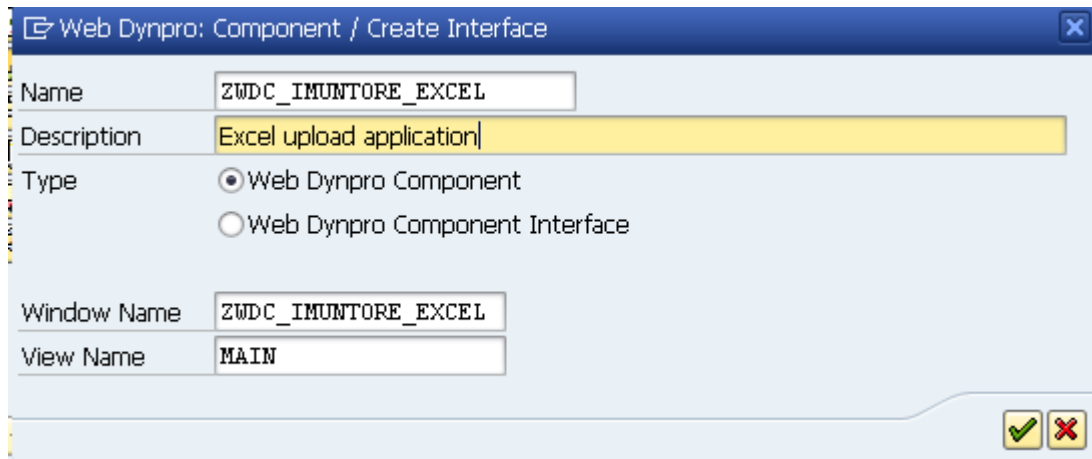
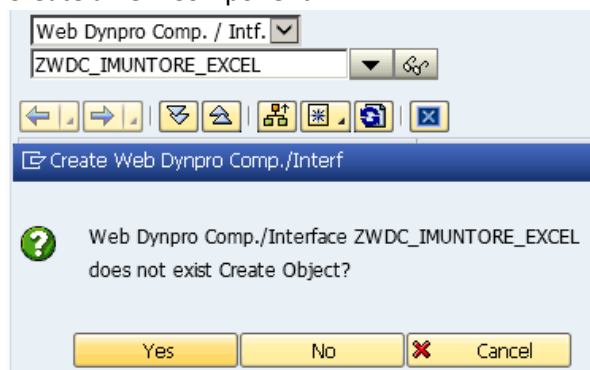


The 'Interface Controller' window shows the 'Context' tab. The title bar includes 'Interface Controller' and 'Active'. The tab bar has 'Properties', 'Context', 'Events', and 'Methods'. Below the tab bar is a toolbar with icons for adding, deleting, and editing context. A table lists context items with columns 'Context', 'Context T...', 'Interfa...', and 'Descr'. The context 'CONTEXT' is selected, and it has a child context 'USER_CONTACT' with a yellow folder icon and a red 'S' icon.



It means that we can use these elements in another WD Components.

2) Create a new Component:



- 3) Go to the WebDynpro Component on the Used Components, put:

Web Dynpro Component: ZWDC_IMUNTORE_EXCEL Active/revised

Description: Excel upload application

Assistance Class:

Created By: IMUNTORE Created On: 08.07.2016

Last Changed By: IMUNTORE Changed On: 08.07.2016

Original Lang.: EN Package: \$TMP

☐ Accessibility Checks Active

Used Components Implemented interfaces

Used Web Dynpro Components

Component Use	Component	Description of Component
OLD_APPLICATION	ZWDC_USER_DISPLAY_IMUNTOR	Display User Contact Information

This is your last ALV WD
Component

- 4) Go to the COMPONENTCONTROLLER:

Component Controller: COMPONENTCONTROLLER Active

Properties Context Attributes Events Methods

Controller Usage

Context COMPONENTCONTROLLER

- CONTEXT

Component Use	Component	View/Controller	Description
	ZWDC_IMUNTORE_EXCEL	ZWDC_IMUNTORE_EXCEL	
OLD_APPLICATION	ZWDC_USER_DISPLAY_IMUNTOR	INTERFACECONTROLLER	Component Controller

Bind the NODE:

Component Controller: COMPONENTCONTROLLER Active(revised)

Properties Context Attributes Events Methods

Controller Usage

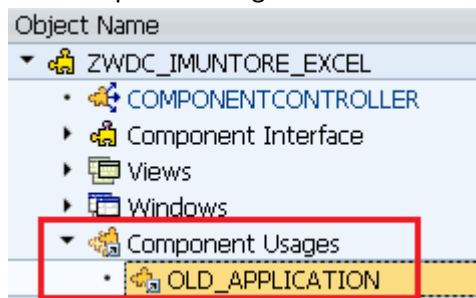
Context COMPONENTCONTROLLER

- CONTEXT
 - USER_CONTACT

Context INTERFACECONTROLLER

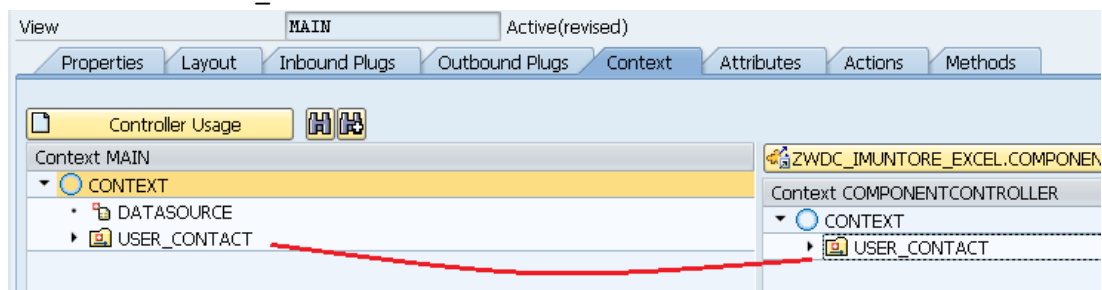
- CONTEXT
 - USER_CONTACT

- 5) The Component Usage will also in the Tree appears:



- 6) Go to the MAIN View

Bind the NODE USER_CONTACT from the CONTROLLER context to the View context:

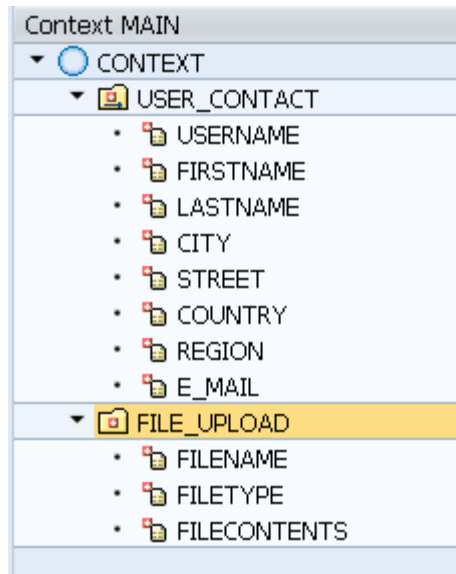


Create a new node:

The 'Create Nodes' dialog box is shown. It has a title bar with a close button. The 'Node Name' field contains 'FILE_UPLOAD'. The 'Interface Node' dropdown is set to 'No'. The 'Input Element (Ext.)' dropdown is set to 'No'. The 'Dictionary structure' field is empty. The 'Cardinality' dropdown is set to '1..n'. The 'Selection' dropdown is set to '0..1'. The 'Init. Lead Selection' dropdown is set to 'Yes'. The 'Singleton' dropdown is set to 'No'. The 'Supply Function' field is empty. At the bottom, there are four buttons: a green checkmark, a green checkmark with the text 'Add Attributes from Structure', a green checkmark with the text 'Additional Node', and a red X.

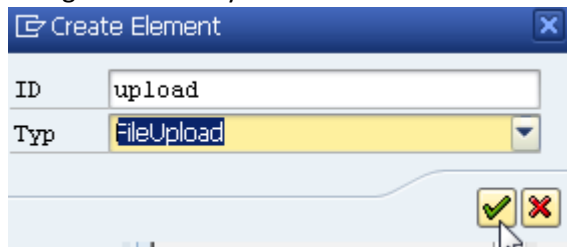
Inside the node create 3 attributes: FILENAME (string), FILETYPE (string), FILECONTENTS (xstring).

You context should look like this:

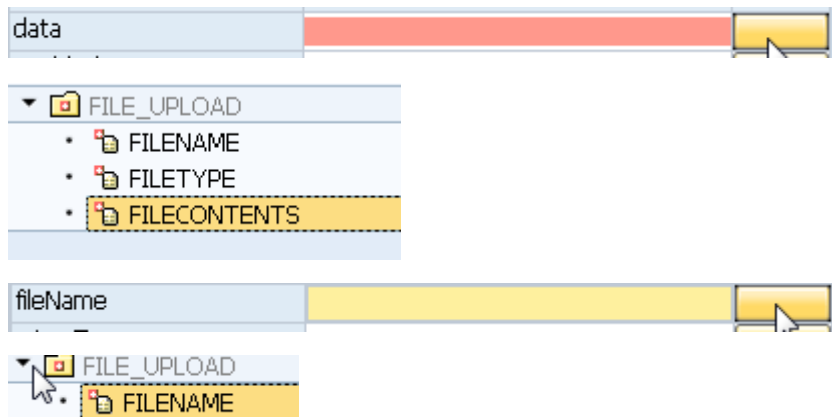


Save and Activate.

Navigate to the Layout tab and create a new element in the Root:



Make its properties exactly like this:



mimeType MAIN.FILE_UPLOAD.FILETYPE



Result:

Property	Value	Binding
Properties (FileUpload)		
ID	UPLOAD	
activateAccessKey	<input type="checkbox"/>	
contextMenuBehaviour	Inherit	
contextMenuId		
data	MAIN.FILE_UPLOAD.FILECONTENTS	
enabled	<input checked="" type="checkbox"/>	
explanation		
fileName	MAIN.FILE_UPLOAD.FILENAME	
mimeType	MAIN.FILE_UPLOAD.FILETYPE	
state	Normal Item	
textDirection	Inherit	
tooltip		
virusScanProfile		
visible	Visible	
width		
Layout Data (FlowData)		
cellDesign	padless	
vGutter	None	

Create also a Button:

Create Element

ID

confirm_upload

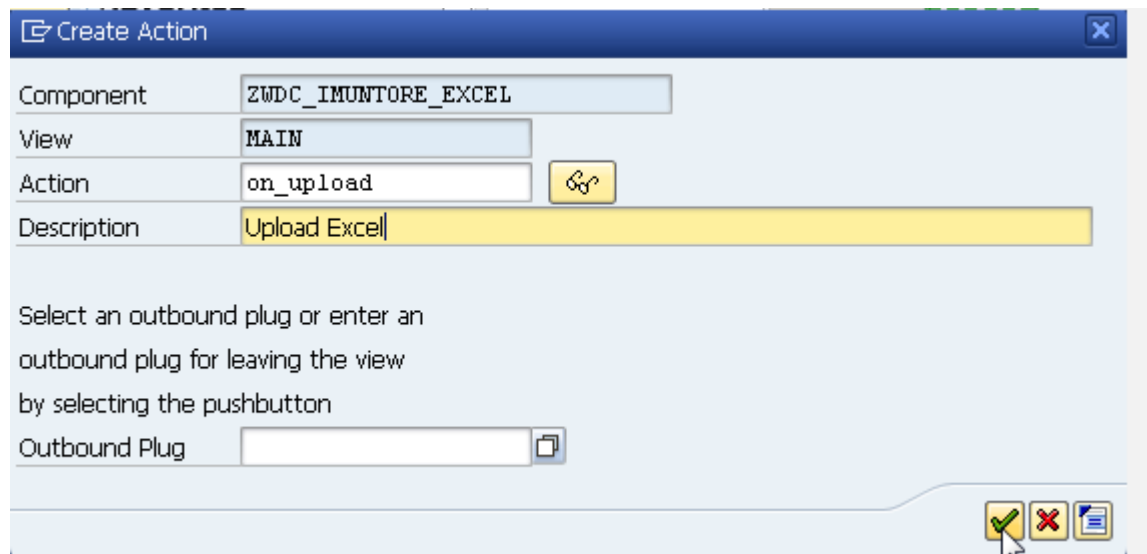
Typ

Button

text

Upload

Create an Action to the Button:

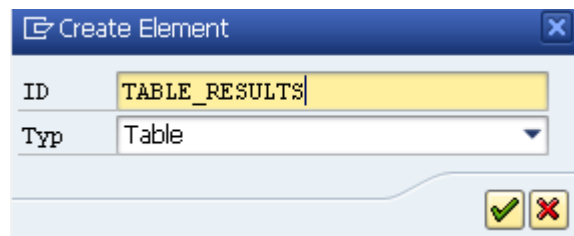


The 'Create Action' dialog box is shown with the following fields:

- Component: ZWDC_IMUNTORE_EXCEL
- View: MAIN
- Action: on_upload
- Description: Upload Excell

Below the fields, there is a text instruction: "Select an outbound plug or enter an outbound plug for leaving the view by selecting the pushbutton". Below this is an 'Outbound Plug' field which is currently empty. At the bottom right, there are three icons: a green checkmark, a red X, and a document icon.

Create another element:

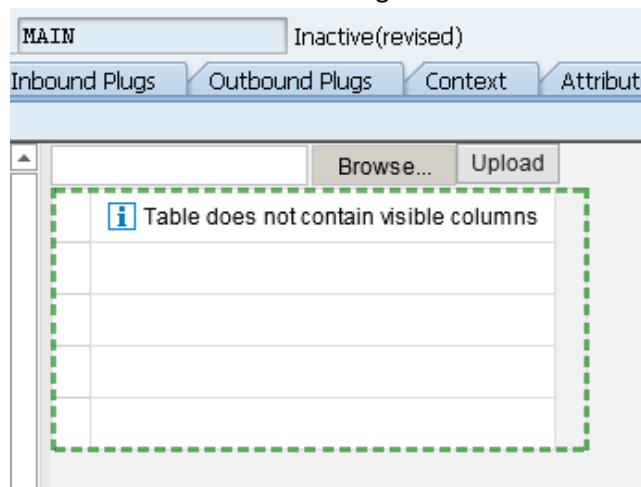


The 'Create Element' dialog box is shown with the following fields:

- ID: TABLE_RESULTS
- Type: Table

At the bottom right, there are two icons: a green checkmark and a red X.

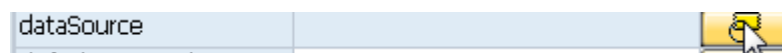
Your screen will look something like this:



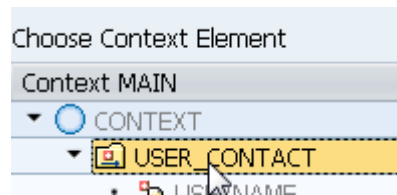
The main application window is shown with the following elements:

- Tab: MAIN
- Status: Inactive(revised)
- Navigation tabs: Inbound Plugs, Outbound Plugs, Context, Attributes
- Buttons: Browse..., Upload
- Table: A table with 5 rows and 1 column. The first row contains an information icon and the text "Table does not contain visible columns".

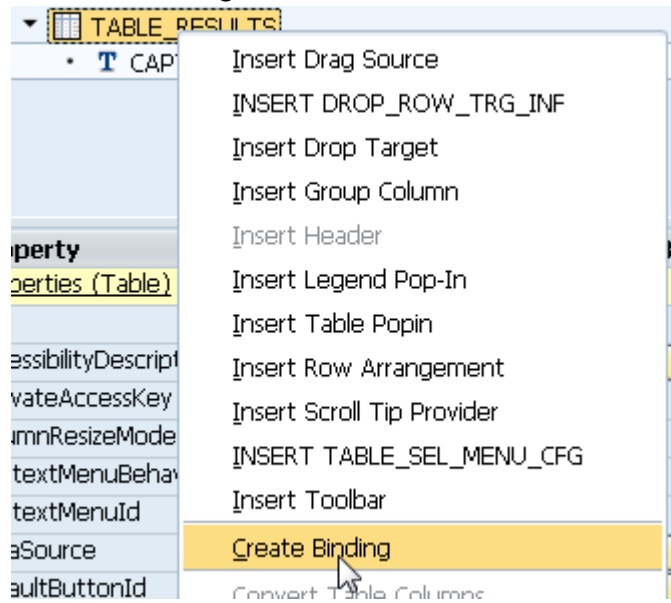
Create a datasource to the table



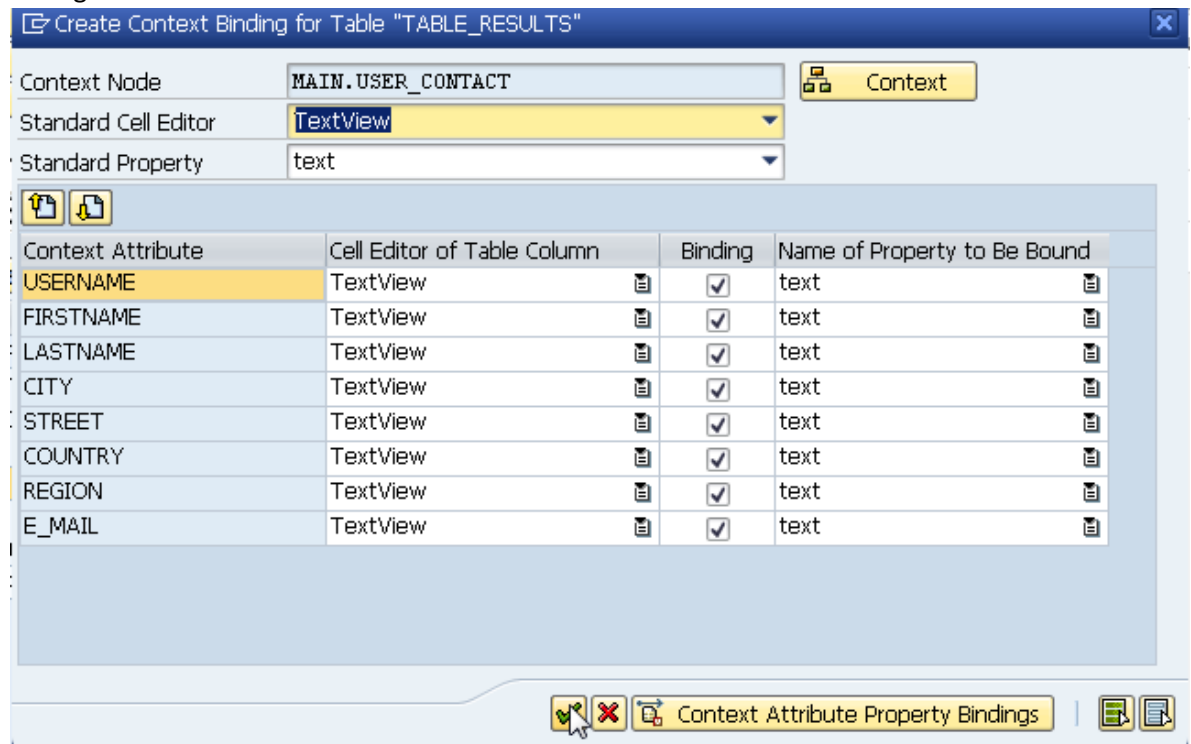
The 'DataSource' configuration is shown with a table that has 2 columns and 1 row. The first column is labeled 'dataSource'. The second column contains a yellow icon with a green checkmark and a red X.



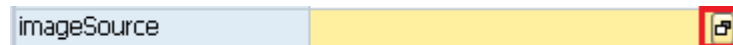
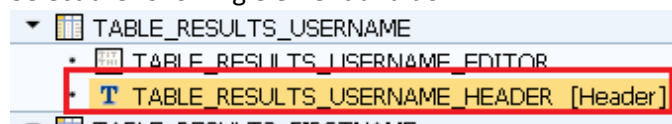
Create the Binding:




Configure like shown below:



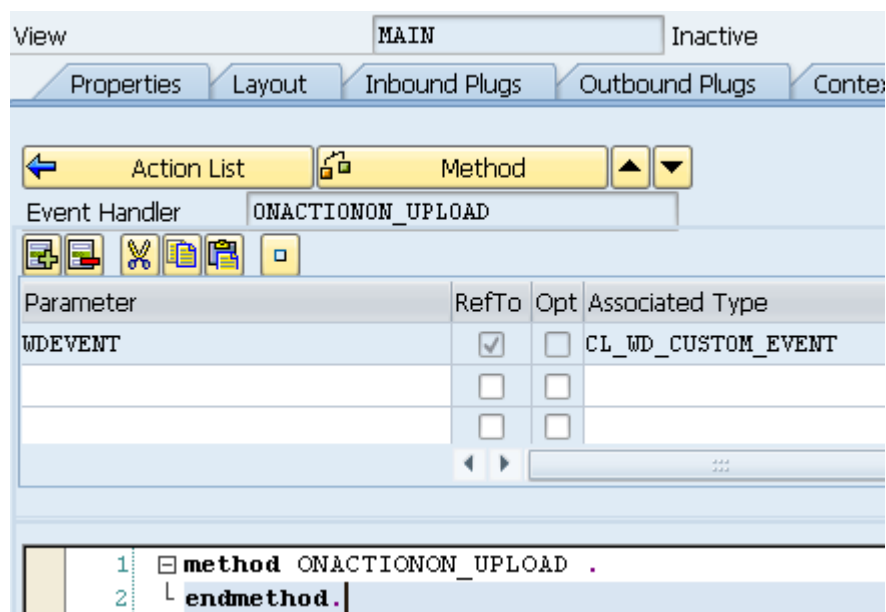
Select the following element and do:



Now your column will be a little more friendly:

	 User	First name
	MAIN.USER_CONTACT.USERNAME	MAIN.USER_CONTACT
	MAIN.USER_CONTACT.USERNAME	MAIN.USER_CONTACT
	MAIN.USER_CONTACT.USERNAME	MAIN.USER_CONTACT
	MAIN.USER_CONTACT.USERNAME	MAIN.USER_CONTACT
	MAIN.USER_CONTACT.USERNAME	MAIN.USER_CONTACT

7) Time to Code!



Copy and Past the following:

It will be explained later by debugging:

```
METHOD onactionon_upload .
```

```
DATA lo_nd_file_upload TYPE REF TO if_wd_context_node.
DATA lo_el_file_upload TYPE REF TO if_wd_context_element.
DATA ls_file_upload TYPE wd_this->element_file_upload.

DATA lo_nd_user_contact TYPE REF TO if_wd_context_node.
DATA lo_el_user_contact TYPE REF TO if_wd_context_element.
DATA ls_user_contact TYPE wd_this->element_user_contact.
```

```

DATA lt_user_contact TYPE TABLE OF wd_this->element_user_contact.

lo_nd_file_upload = wd_context->get_child_node( name = wd_this-
>wdctx_file_upload ).
lo_el_file_upload = lo_nd_file_upload->get_element( ).
* Get all declared attributes
lo_el_file_upload->get_static_attributes(
  IMPORTING
    static_attributes = ls_file_upload ).

* Create object of class to read .xlsx file contents
DATA(lref_excel) = NEW cl_fdt_xl_spreadsheet( document_name = ls_file_u
pload-filename
                                           xdocument      = ls_file_u
pload-filecontents ).

lref_excel-
>if_fdt_doc_spreadsheet~get_worksheet_names( IMPORTING worksheet_names =
DATA(lt_worksheets) ).
* Condition to check whether .xlsx file has any active worksheets
READ TABLE lt_worksheets INDEX 1 INTO DATA(lv_name).
IF sy-subrc = 0.
* Get reference of .xlsx file contents in the active worksheet
DATA(lref_data) = lref_excel-
>if_fdt_doc_spreadsheet~get_itab_from_worksheet( lv_name ).
ENDIF.

FIELD-SYMBOLS: <fs_table> TYPE ANY TABLE.
* Fetch all records in the active worksheet
ASSIGN lref_data->* TO <fs_table>.
IF <fs_table> IS ASSIGNED.

  LOOP AT <fs_table> ASSIGNING FIELD-SYMBOL(<fs_l_table>).

    ASSIGN COMPONENT 'A' OF STRUCTURE <fs_l_table> TO FIELD-
SYMBOL(<fs_value>).
    IF <fs_value> IS ASSIGNED.
      ls_user_contact-username = <fs_value>.
    ENDIF.
    UNASSIGN <fs_value>.
    ASSIGN COMPONENT 'B' OF STRUCTURE <fs_l_table> TO <fs_value>.
    IF <fs_value> IS ASSIGNED .
      ls_user_contact-firstname = <fs_value>.
    ENDIF.
    UNASSIGN <fs_value>.
    ASSIGN COMPONENT 'C' OF STRUCTURE <fs_l_table> TO <fs_value>.
    IF <fs_value> IS ASSIGNED.
      ls_user_contact-lastname = <fs_value>.
    ENDIF.
    UNASSIGN <fs_value>.
    ASSIGN COMPONENT 'D' OF STRUCTURE <fs_l_table> TO <fs_value>.
    IF <fs_value> IS ASSIGNED.
      ls_user_contact-city = <fs_value>.

```

```

ENDIF.
UNASSIGN <fs_value>.
ASSIGN COMPONENT 'E' OF STRUCTURE <fs_l_table> TO <fs_value>.
IF <fs_value> IS ASSIGNED.
    ls_user_contact-street = <fs_value>.
ENDIF.
UNASSIGN <fs_value>.
ASSIGN COMPONENT 'F' OF STRUCTURE <fs_l_table> TO <fs_value>.
IF <fs_value> IS ASSIGNED.
    ls_user_contact-country = <fs_value>.
ENDIF.
UNASSIGN <fs_value>.
ASSIGN COMPONENT 'G' OF STRUCTURE <fs_l_table> TO <fs_value>.
IF <fs_value> IS ASSIGNED.
    ls_user_contact-region = <fs_value>.
ENDIF.
UNASSIGN <fs_value>.
ASSIGN COMPONENT 'H' OF STRUCTURE <fs_l_table> TO <fs_value>.
IF <fs_value> IS ASSIGNED.
    ls_user_contact-e_mail = <fs_value>.
ENDIF.
UNASSIGN <fs_value>.
APPEND ls_user_contact TO lt_user_contact.
CLEAR ls_user_contact.
ENDLOOP.

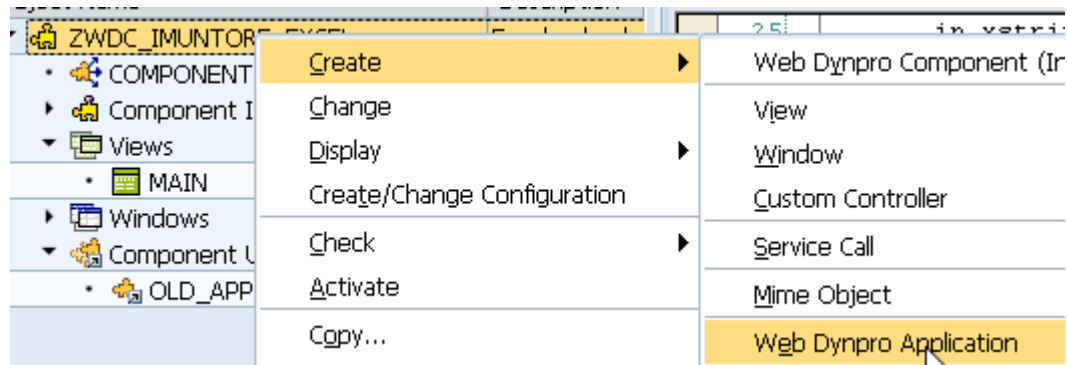
lo_nd_user_contact = wd_context->get_child_node( name = wd_this-
>wdctx_user_contact ).
lo_el_user_contact = lo_nd_user_contact->get_element( ).
lo_nd_user_contact->bind_table( new_items = lt_user_contact ).

ENDIF.

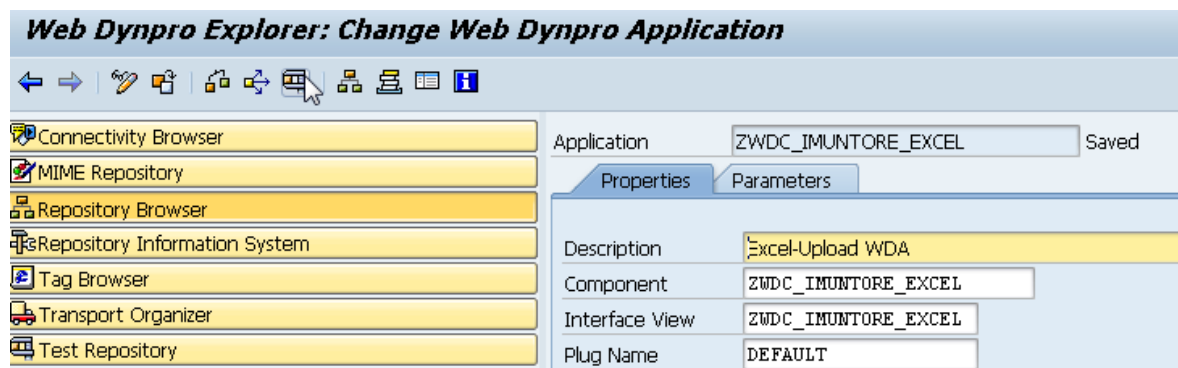
ENDMETHOD.

```

8) Create the WDA:



Execute it.



Your application will look like this:

	First name	User	Last name	City	Street	Country	Region	E-Mail Address

Now upload the following file:



Users_Excel_Contacts_3.xlsx

You will get the following error:





500 SAP Internal Server Error

ERROR: Number of elements in the collection of node **COMPONENTCONTROLLER.1.USER_CONTACT** violates the cardinality (termination: RABAX_STATE)

And this is the reason:

Component Controller COMPONENTCONTROLLER Active

Properties Context Attributes Events Methods

Controller Usage  

Context COMPONENTCONTROLLER

- CONTEXT
 - USER_CONTACT
 - USERNAME
 - FIRSTNAME
 - LASTNAME
 - CITY
 - STREET
 - COUNTRY
 - REGION
 - E_MAIL

Context INTE

CONTEXT

USE

Property	Value
Nodes	
Node Name	USER_CONTACT
Interface Node	
Input Element (Ext.)	
Dictionary structure	
Cardinality	1..1
Selection	0..1

Since it was inherited from the old component. Open the old component and do the following changes:

Controller for ZWDC_USER_DISPLAY_IMUNTOR

Pattern Pretty Printer Controller Documentation Switch Context Editor View

Component Controller COMPONENTCONTROLLER Active

Properties Context Attributes Events Methods

Controller Usage

Context COMPONENTCONTROLLER

- CONTEXT
 - USER_CONTACT
 - USERNAME
 - FIRSTNAME
 - LASTNAME
 - CITY
 - STREET
 - COUNTRY
 - REGION
 - E_MAIL

ZWDC_USER_DISPLAY_IMUNTOR

Context W_MAIN

- CONTEXT

Property	Value
Nodes	
Node Name	USER_CONTACT
Interface Node	
Input Element (Ext.)	
Dictionary structure	
Cardinality	1..n

Save and activate!

Go to our new Component and check out the results

Controller for ZWDC_IMUNTORE_EXCEL

Pattern Pretty Printer Controller Documentation Switch Context Editor View

Component Controller COMPONENTCONTROLLER Inactive

Properties Context Attributes Events Methods

Controller Usage

Context COMPONENTCONTROLLER

- CONTEXT
 - USER_CONTACT
 - USERNAME
 - FIRSTNAME
 - LASTNAME
 - CITY
 - STREET
 - COUNTRY
 - REGION
 - E_MAIL

Context INTERFACECONTROLLER

- CONTEXT
 - USER_CONTACT

Property	Value
<u>Nodes</u>	
Node Name	USER_CONTACT
Interface Node	
Input Element (Ext.)	
Dictionary structure	
Cardinality	1..n
Selection	0..1

Save and activate all.

Run the WDA again uploading the file.

Final Result:

		<input type="button" value="Browse..."/>	<input type="button" value="Upload"/>					
	First name	User	Last name	City	Street	Country	Region	E-Mail Address
	Igor	IMUNTORE	Muntoreanu	Blumenau	Ingo Hering	BR	BR	igor.muntorea@t-systems.com.br
	Fernando	FBUGMANN	Bugmann	Blumenau	Ingo Hering	BR	BR	fernando.bugmann@t-systems.com.br
	Chuan	CHUANG	Huang	Darmstadt	Pfnorstraße	DE	DE	Chuan.Huang@t-systems.com

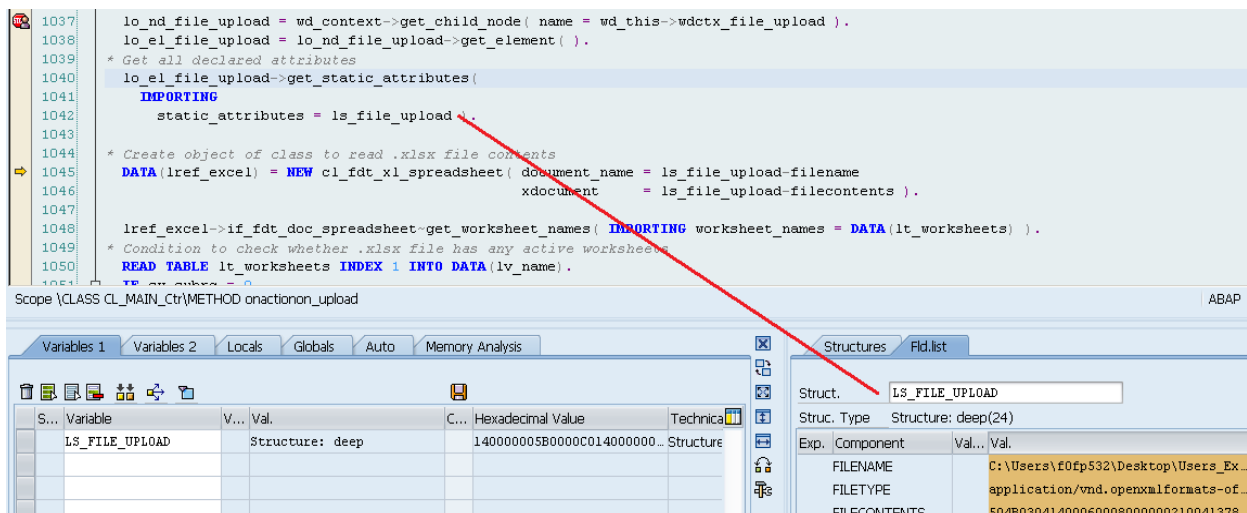
Understanding the code by debugging it:

Put an external break-point on the following line:

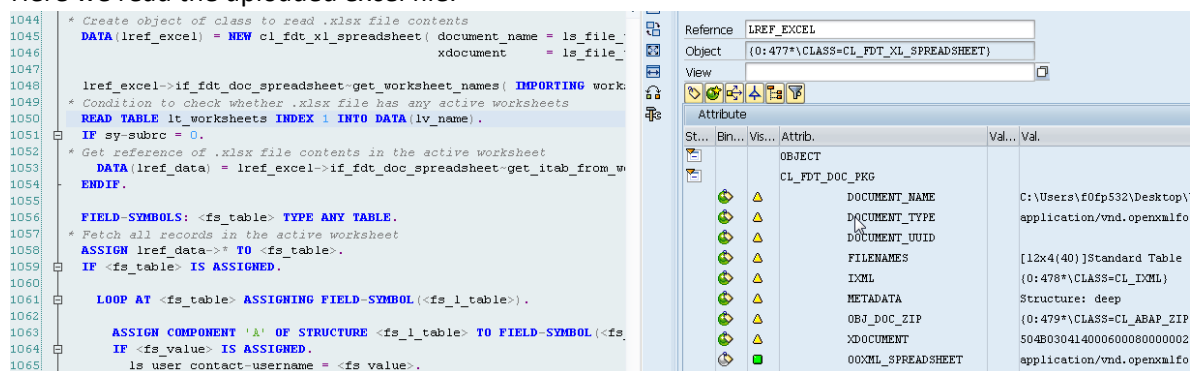
```
1  METHOD onactionon_upload .
2
3      DATA lo_nd_file_upload TYPE REF TO
4      DATA lo_el_file_upload TYPE REF TO
5      DATA ls_file_upload TYPE wd_this->e
6
7      DATA lo_nd_user_contact TYPE REF TO
8      DATA lo_el_user_contact TYPE REF TO
9      DATA ls_user_contact TYPE wd_this->
10     DATA lt_user_contact TYPE TABLE OF
11
12     lo_nd_file_upload = wd_context->get
```

Execute the WDA and upload the file:

Until this point, we are just reading the FILE_UPLOAD node.

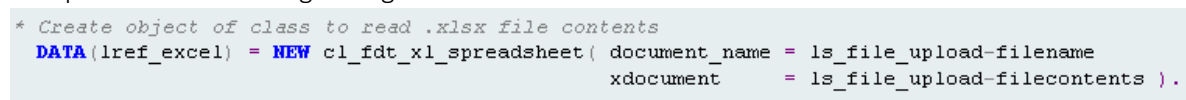


Here we read the uploaded excel file:



All the details are inside lref_excel.

This part is ABAP 7.4 Programming:



It is just like doing that below on old ABAP versions:

CREATE OBJECT lref_excel

Importing

```

document_name = ls_file_upload-filename
xdocument      = ls_file_upload-filecontents

```

This part of the code means that the program is reading the Plan1 worksheet:

```

1048 | lref_excel->if_fdt_doc_spreadsheet-get_worksheet_names( IMPORTING worksheet_names = DATA(lt_worksheets) ).
1049 | * Condition to check whether .xlsx file has any active worksheets
1050 | READ TABLE lt_worksheets INDEX 1 INTO DATA(lv_name).
1051 | IF sy-subrc = 0.

```

Field	LV_NAME
Data Type	CString(5)
Absolute Type	\TYPE=STRING
View	Fast Display
Plan1	50006C0061006E003100

	A	B	C	D	E	F	G	H
1	IMUNTORE	Igor	Muntoreanu	Blumenau	Ingo Hering	BR	BR	igor.muntorea@t-systems.com.br
2	FBUGMANN	Fernando	Bugmann	Blumenau	Ingo Hering	BR	BR	fernando.bugmann@t-systems.com.br
3	CHUANG	Chuan	Huang	Darmstadt	Pfnorstraße	DE	DE	Chuan.Huang@t-systems.com

The <fs_table> contains all the contents of the EXCEL file

```

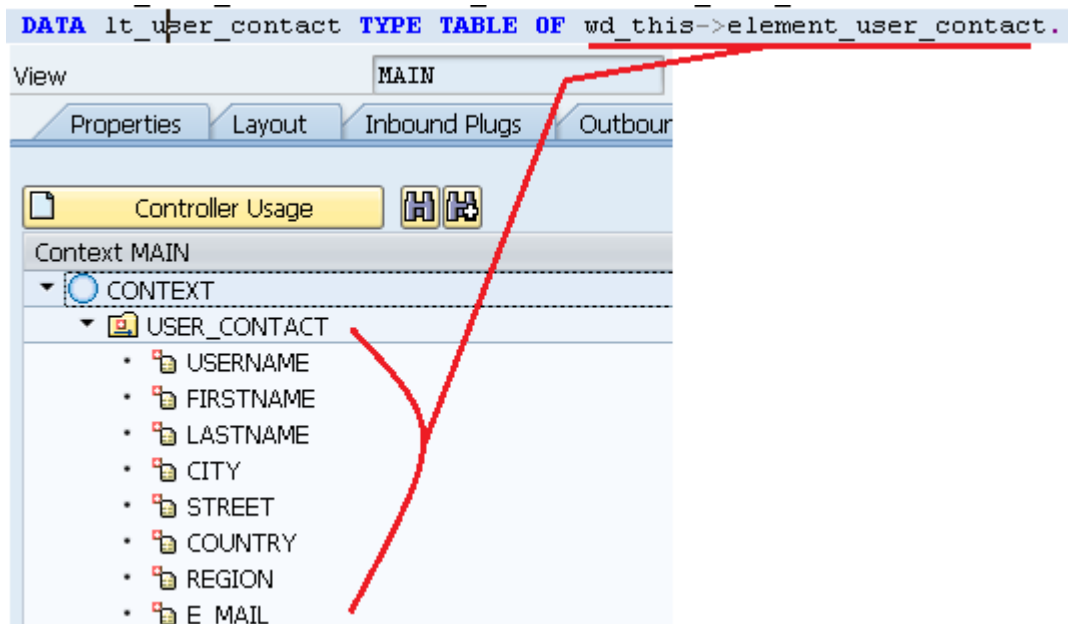
50 | READ TABLE lt_worksheets INDEX 1 INTO DATA(lv_name).
51 | IF sy-subrc = 0.
52 | * Get reference of .xlsx file contents in the active worksheet
53 | DATA(lref_data) = lref_excel->if_fdt_doc_spreadsheet-get_itab_from_w
54 | ENDF.
55 |
56 | FIELD-SYMBOLS: <fs_table> TYPE ANY TABLE.
57 | * Fetch all records in the active worksheet.
58 | ASSIGN lref_data->* TO <fs_table>.
59 | IF <fs_table> IS ASSIGNED.
60 |
61 | LOOP AT <fs_table> ASSIGNING FIELD-SYMBOL(<fs_l_table>).

```

S...	Variable	V...
	<FS_TABLE>	
	LREF_DATA	

Row	A [CString]	B [CString]	C [CString]	D [CString]	E [CString]	F [CString]	G [CString]	H [CString]
1	IMUNTORE	Igor	Muntoreanu	Blumenau	Ingo Hering	BR	BR	igor.muntorea@t-systems.com.br
2	FBUGMANN	Fernando	Bugmann	Blumenau	Ingo Hering	BR	BR	fernando.bugmann@t-systems.com.
3	CHUANG	Chuan	Huang	Darmstadt	Pfnorstraße	DE	DE	Chuan.Huang@t-systems.com

The following loop is to feed the final internal table lt_user_contact that is the type of the context node that we built:



And finally here we set the final fed internal table to the context and the layout table:

```
1107 | lo_nd_user_contact = wd_context->get_child_node( name = wd_this->wdctx_user_contact ).  
1108 | lo_el_user_contact = lo_nd_user_contact->get_element( ).  
1109 | lo_nd_user_contact->bind_table( new_items = lt_user_contact ).  
1110 |
```

9) Homework

- Create a Download button to do the reverse logic:
- Validate the File entered. Check if it is XLS or XLSX and if now send an error message.

